

**RANCANG BANGUN *VIEW CONTROLLER* MENGGUNAKAN
SENSOR AKSELEROMETER PADA GAME BUS SIMULATOR
SEBAGAI SARANA *USER EXPERIENCE (UX)* BERBASIS SISTEM
*EMBEDDED***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Bukhori Darmawan
NIM: 145150300111020



PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

RANCANG BANGUN VIEW CONTROLLER MENGGUNAKAN SENSOR
AKSELEROMETER PADA GAME BUS SIMULATOR SEBAGAI SARANA *USER*
EXPERIENCE (UX) BERBASIS SISTEM *EMBEDDED*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :

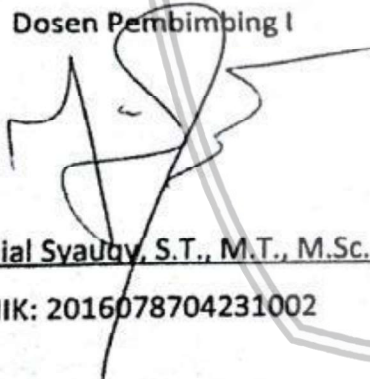
Bukhori Darmawan

NIM: 145150300111020

Skripsi ini telah diuji dan dinyatakan lulus pada
27 Desember 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Dahnial Syahby, S.T., M.T., M.Sc.

NIK: 2016078704231002

Dosen Pembimbing II



Mochammad Hannats Hanafi Ichsan, S.ST, M.T

NIK: 201405 881229 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoro-kurpiawan, S.T., M.T., Ph.D.

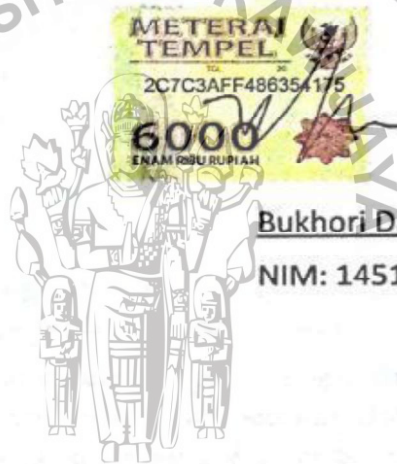
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 3 Januari 2019



Bukhori Darmawan

NIM: 145150300111020

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena Rahmat-Nya lah penulis dapat menyelesaikan skripsi ini. Adapun maksud penyusunan skripsi ini adalah untuk memenuhi salah satu persyaratan dalam menempuh ujian Sarjana Fakultas Ilmu Komputer. Judul skripsi yang disusun adalah: **"RANCANG BANGUN VIEW CONTROLLER MENGGUNAKAN SENSOR AKSELEROMETER PADA GAME BUS SIMULATOR SEBAGAI SARANA USER EXPERIENCE (UX) BERBASIS SISTEM EMBEDDED"**.

Dalam menyusun skripsi ini ada banyak kesulitan dan hambatan yang dialami oleh peneliti, tetapi semua itu telah dapat diatasi dengan baik berkat dukungan dan bantuan dari berbagai pihak. Untuk itulah pada kesempatan ini peneliti mengucapkan terima kasih kepada:

1. Bapak dan Ibu serta keluarga besar peneliti yang telah memberikan dukungan, semangat serta doanya hingga terselesaikannya skripsi ini.
2. Bapak Dahnia Syauqy, S.T., M.T., M.Sc. selaku dosen pembimbing I yang telah memberikan pengarahan dan bimbingannya kepada peneliti selama proses pengerjaan skripsi sehingga dapat terselesaikan dengan baik.
3. Bapak Mochammad Hannats Hanafi Ichsan, S.ST, M.T selaku dosen pembimbing II yang telah memberikan pengarahan dan bimbingannya selama proses pembuatan laporan skripsi sehingga skripsi ini dapat terselesaikan dengan baik.
4. Teman-teman remaja masjid klaseman yang senantiasa memberikan semangat dan motivasi selama proses pengerjaan skripsi ini.
5. Seluruh pimpinan dekanat beserta jajarannya, seluruh civitas akademika, serta seluruh teman-teman dari keminatan Teknik Komputer angkatan 2014 Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak membantu peneliti dalam proses pengerjaan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, oleh karena itu untuk segala kritik dan saran yang membangun penulis ucapkan terima kasih. Penulis mengharapkan semoga skripsi ini dapat berguna bagi yang membutuhkannya.

Malang, 3 Januari 2019

Penulis

darmawanbukhori@gmail.com

ABSTRAK

Berkembangnya teknologi pada khususnya dibidang game membuat manusia menginginkan bagaimana cara bermain game dengan pengalaman yang berbeda atau bisa dikatakan merasakan *experience* dalam bermain game. Pada game berjenis *racing* banyak sekali kesempatan untuk menerapkan *user experience*, pada kesempatan ini peneliti mengembangkan suatu sistem untuk menggantikan sebuah *mouse* dalam bermain game berjenis *racing*. Berdasarkan itu dibutuhkan sistem *view controller*, implementasi sistem *view controller* dilakukan dengan perangkat arduino pro mikro sebagai mikorkontroler dan sensor MPU9250 sebagai input gerakan yang kemudian dihubungkan ke laptop dengan media komunikasi berupa kabel serial dan terpasang pada kepala *user* dengan *headband* secara *embedded*. Untuk menghasilkan *user experience* yang berbeda dalam bermain game, sistem mampu membatasi pergerakan *view controller* dengan batasan 40° untuk pergerakan kepala *user* pada saat menoleh ke kanan dan ke kiri sehingga *user* dapat leluasa ketika menoleh ke kanan dan ke kiri sesuai keinginan *user*. Sistem akan diuji secara keseluruhan dari beberapa sudut yang akan menghasilkan nilai sudut dengan nilai selisih rata-rata RMSE yang rendah dimana sistem sudah dapat digunakan dan diuji dengan metode survei deskriptif kuantitatif dari 30 korespondensi untuk mendapatkan nilai-nilai kategori dari *user experience*. Hasil dari pengujian didapatkan rata-rata pengujian sudut sensor adalah kurang lebih 0.2° dan hasil pengujian korespondensi *user experience* sistem mampu mendapatkan rata-rata kategori baik berdasarkan *feedback* yang diberikan oleh 30 *user*.

Kata kunci: MPU9250, game, *view controller*, *user experience*.

ABSTRACT

The development of technology in particular in the field of games makes people want how to play games with a different *experience* or could be said to feel the *experience* in playing the game. -*Racing* game on an awful lot of opportunities to apply *user experience*, on this occasion the researchers to develop a sistem to replace a *mouse* in-game play *racing*. Based on that required a sistem *view* controller, sistem implementation *view* controller is done with the arduino pro micro as mikorkontroler and MPU9250 as input motion sensors are then connected to a laptop with media communications either a serial cable and installed on the *user's* head with a headband are *embedded*. To generate a different *user experience* in playing a game, the sistem is able to restrict the movement of the *view* controller with 40 ° restrictions to the movement of the *user's* head on when turned to the right and to the left so the *user* can freely when turned to the right and to the the left will be used. The sistem will be tested in its entirety from some corners that will result in the value of the angle with the value of the difference in average low RMSE where the sistem can already be used and tested by quantitative descriptive survei method of correspondence for 30 get the category values from the *user experience*. The results of the test are obtained by averaging the test angle sensor is approximately 0.2 ° and the corresponding *user experience* testing results sistems were able to get a good category average based on feedback given by 30 *users*.

keyword: MPU9250, game, *view* controller, *user experience*.

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori.....	6
2.2.1 <i>User Experience</i>	6
2.2.2 Game Bus Simulator	7
2.2.3 Arduino Pro Mikro.....	8
2.2.4 MPU9250 GY-91	9
2.2.5 Inter Integrated Circuit Bus (I2C).....	11
2.2.6 Arduino IDE.....	11
2.2.7 <i>Library Mouse.h</i>	12
2.2.8 <i>Library IMU MPU6050.h</i>	12
2.2.9 <i>Root Mean Square Error</i>	12
2.2.10 Skala Likert.....	12
BAB 3 METODOLOGI.....	14
3.1 Identifikasi Masalah	15
3.2 Studi Literatur	15
3.3 Analisis Kebutuhan	15
3.3.1 Kebutuhan Perangkat Keras	15
3.3.2 Kebutuhan Perangkat Lunak.....	16
3.4 Perancangan Sistem.....	16
3.5 Implementasi	17
3.6 Pengujian dan Analisis	17
3.6.1 Pengujian nilai sudut perangkat <i>view controller</i>	17
3.6.2 Pengujian fungsional dari <i>view controller</i>	17
3.6.3 Pengujian <i>user</i> yang akan di uji coba dalam penggunaan <i>view controller</i> dengan korespondensi kuisioner	17

	3.7 Penarikan Kesimpulan.....	17
BAB 4	REKAYASA KEBUTUHAN	18
	4.1 Gambaran Umum Sistem.....	18
	4.2 Karakteristik Pengguna	18
	4.3 Analisis Kebutuhan Sistem.....	19
	4.3.1 Kebutuhan Fungsional	19
	4.3.2 Kebutuhan Non Fungsional.....	19
	4.3.2.1 Kebutuhan Perangkat keras.....	20
	4.3.2.2 Kebutuhan Perangkat lunak	20
	4.4 Batasan Desain Sistem	21
BAB 5	PERANCANGAN DAN IMPLEMENTASI.....	22
	5.1 Perancangan Sistem.....	22
	5.1.1 Diagram Blok Sistem	22
	5.1.2 Perancangan alat dan peletakan <i>view controller</i>	23
	5.2 Perancangan Perangkat Keras	23
	5.3 Perancangan Perangkat Lunak.....	25
	5.3.1 Perancangan kode program utama.....	25
	5.3.2 Perancangan kode program konversi <i>yaw, pitch</i> dan <i>roll</i>	27
	5.3.3 Perancangan kendali kursor	28
	5.4 Perancangan Prosedur Penggunaan Aplikasi pada Sistem.....	29
	5.4.1 Perancangan prosedur penggunaan game bus simulator untuk <i>user</i>	29
	5.4.2 Perancangan prosedur penggunaan aplikasi stream dari laptop ke <i>smartphone</i>	31
	5.5 Implementasi Sistem.....	32
	5.5.1 Implementasi alat dan peletakan <i>view controller</i>	32
	5.5.2 Implementasi perangkat keras	33
	5.5.3 Implementasi perangkat lunak	34
	5.5.4 Implementasi prosedur penggunaan aplikasi.....	36
BAB 6	PENGUJIAN DAN ANALISIS	40
	6.1 Pengujian Sensor MPU9250.....	40
	6.1.1 Tujuan Pengujian	40
	6.1.2 Prosedur Pengujian.....	40
	6.1.3 Pelaksanaan pengujian	40
	6.1.4 Hasil Pengujian.....	40
	6.1.5 Analisis Hasil Pengujian.....	42
	6.2 Pengujian Fungsional Sistem	42
	6.2.1 Tujuan Pengujian	42
	6.2.2 Prosedur Pengujian.....	42
	6.2.3 Pelaksanaan pengujian	43
	6.2.4 Hasil Pengujian.....	43
	6.2.5 Analisis Hasil Pengujian.....	45
	6.3 Pengujian Korespondensi	45
	6.3.1 Tujuan Pengujian	45

6.3.2	Prosedur Pengujian.....	45
6.3.3	Pelaksanaan pengujian	45
6.3.4	Hasil Pengujian.....	46
6.3.5	Analisis Hasil Pengujian.....	51
BAB 7	KESIMPULAN	
7.1	Kesimpulan.....	56
7.2	Saran	56
DAFTAR PUSTAKA		
LAMPIRAN		



DAFTAR TABEL

Tabel 2.1 Penelitian Terkait.....	5
Tabel 2.1 Penelitian Terkait (lanjutan)	6
Tabel 2.2 Rekomendasi Sistem Game Bus Simulator.....	8
Tabel 2.3 Karakteristik Aeduino pro mikro	9
Tabel 2.4 Karakteristik MPU9250.....	10
Tabel 2.4 Karakteristik MPU 9250 (Lanjutan)	11
Tabel 5.1 Koneksi pin perancangan perangkat keras sistem	21
Tabel 5.2 Mendefinisikan <i>library</i>	34
Tabel 5.3 Mendeklarasikan objek, menginialisasi fungsi DMP	34
Tabel 5.4 Kode program untuk mendapatkan nilai dari sensor	35
Tabel 5.5 Kode program inialisasi <i>library mouse</i>	35
Tabel 5.6 Kode program untuk mendapatkan konversi data dan menggerakkan <i>mouse</i>	36
Tabel 6.1 Hasil pengujian pada sumbu X	40
Tabel 6.2 Hasil pengujian pada sumbu Y	41
Tabel 6.3 Hasil pengujian pada sumbu Y	41
Tabel 6.4 Hasil rekapitulasi RMSE	42
Tabel 6.5 Hasil pengujian ketika menoleh ke kanan.....	43
Tabel 6.6 Hasil pengujian ketika menoleh ke kiri.....	44
Tabel 6.7 Hasil pengujian koresponden.....	53

DAFTAR GAMBAR

Gambar 2.1	Game Bus Simulator	7
Gambar 2.2	First person mode.....	8
Gambar 2.3	Diagram komponen Arduino Pro Mikro	9
Gambar 2.4	MPU9250 GY-91	10
Gambar 2.5	Pitch, roll dan yaw	11
Gambar 2.6	(a) Diagram blok 12C (b) Start dan stop sequence 12C	12
Gambar 3.1	Alur Metodologi.....	14
Gambar 3.2	Perancangan <i>View Controller</i> menggunakan Sensor MPU9250 ...	16
Gambar 3.3	Diagram blok perangkat keras.....	16
Gambar 5.1	Diagram blok sistem <i>View Controller</i>	22
Gambar 5.2	Ilustrasi perancangan alat tampak samping pada sistem <i>view controller</i>	23
Gambar 5.3	Skema perencanaan perangkat keras sistem <i>view controller</i>	23
Gambar 5.4	Konfigurasi sambungan SCL dan SDA	24
Gambar 5.5	Diagram alir kode program utama	25
Gambar 5.6	Diagram alir kode program konversi yaw, pitch dan roll	27
Gambar 5.7	Diagram alir kode program <i>library mouse</i>	28
Gambar 5.8	(a) penggunaan sistem pada saat menghadap ke depan (b) ilustrasi penggunaan game bus simulator ketika menghadap ke depan.....	29
Gambar 5.9	(a) penggunaan sistem pada saat menghadap ke kanan (b) ilustrasi penggunaan game bus simulator ketika menghadap ke kanan	30
Gambar 5.10	(a) penggunaan sistem pada saat menghadap ke kiri (b) ilustrasi penggunaan game bus simulator ketika menghadap ke kiri	30
Gambar 5.11	Aplikasi Trinus VR	31
Gambar 5.12	Penggunaan aplikasi Trinus VR.....	32
Gambar 5.13	Pemakaian perangkat <i>view controller</i> tampak depan (b) pemakaian perangkat <i>view controller</i> tampak depan	33
Gambar 5.14	<i>view controller</i> tampak depan, samping, dan atas	33
Gambar 5.15	(a) Aplikasi trinus VR pada laptop (b) Aplikasi trinus VR pada <i>smartphone</i>	37
Gambar 5.16	Aplikasi trinus VR pada <i>smartphone</i> yang sudah terkoneksi	37
Gambar 5.17	(a) <i>user</i> menggunakan <i>view controller</i> dalam posisi melihat ke depan (b) posisi didalam game bus simulator	38
Gambar 5.18	(a) <i>user</i> menggunakan <i>view controller</i> dalam posisi melihat ke kanan (b) posisi didalam game bus simulator	38
Gambar 5.19	(a) <i>user</i> menggunakan <i>view controller</i> dalam posisi melihat ke kiri (b) posisi didalam game bus simulator.....	39
Gambar 6.1	Grafik hasil pengujian menoleh ke kanan pada sudut 45 derajat...	43
Gambar 6.2	Grafik hasil pengujian menoleh ke kiri pada sudut 45 derajat	44

Gambar 6.3	diagram hasil umpan balik kesesuaian menggunakan <i>view</i> controller	46
Gambar 6.4	diagram hasil umpan balik kesesuaian menggunakan <i>view</i> controller	46
Gambar 6.5	diagram hasil umpan balik kesesuaian menggunakan <i>view</i> controller	47
Gambar 6.6	diagram hasil umpan balik nilai menggunakan <i>view</i> controller	47
Gambar 6.7	diagram hasil umpan balik nilai menggunakan <i>view</i> controller	48
Gambar 6.8	diagram hasil umpan balik nilai menggunakan <i>view</i> controller	48
Gambar 6.9	diagram hasil umpan balik kemudahan menggunakan <i>view</i> controller	49
Gambar 6.10	diagram hasil umpan balik kemudahan menggunakan <i>view</i> controller	49
Gambar 6.11	diagram hasil umpan balik kemudahan menggunakan <i>view</i> controller	50
Gambar 6.12	diagram hasil umpan balik perasaan menggunakan <i>view</i> controller	50
Gambar 6.13	diagram hasil umpan balik perasaan menggunakan <i>view</i> controller	51
Gambar 6.14	diagram hasil umpan balik perasaan menggunakan <i>view</i> controller	51

DAFTAR LAMPIRAN

Lampiran 1. Lampiran koresponden	59
Lampiran 2. Lampiran koresponden	60



BAB 1 PENDAHULUAN

1.1 Latar belakang

Saat ini, Bermain game merupakan sebuah aktifitas yang dapat membantu kesehatan mental dan kemampuan berfikir otak, bermain game dapat mengurangi tingkat depresi pada pikiran seseorang serta melatih kesabaran dan melatih kesabaran. Dengan bermain game dapat mengisi waktu senggang serta bisa bersenang-senang dan berolahraga ringan. Bermain game dapat dinikmati dengan bermain sendiri ataupun bermain secara bersama dengan teman-teman (Purwoko, 2018). Bermain game berjenis *Racing* dengan judul game Bus Simulator adalah sebuah kegiatan bermain game yang dapat menambah *User Experience* (UX) dengan *adrenalin* yang berbeda.

Game Bus Simulator adalah game berjenis *Racing* dimana pengguna dapat merasakan sensasi bermain game dengan selayaknya seperti sedang menyetir Bus. Game Bus Simulator dikembangkan oleh Stillalive Studios dan dipublikasikan oleh Astragon Entertainment Gmbh pada tahun 2016 (steam, 2016). Game ini memiliki misi yang cukup menantang, *user* dapat mengatur target sendiri dan berusaha mencapainya untuk mendapatkan *reward*. Untuk membuka seluruh peta dan memiliki rute di setiap distrik, *user* harus menyelesaikan satu per satu misi. Bentuk misinya juga berbagai macam sehingga *user* dapat membeli Bus tertentu untuk menjalankan misi. Game ini menawarkan 2 macam *view* sudut pandang yaitu *long view* dan *first person view*. Dalam mode *first person view* pengguna berada didalam *dashboard* bus sehingga pengguna dapat merasakan *User Experience* (UX) selayaknya berada di dalam *dashboard* bus sesungguhnya.

User Experience (UX) adalah evaluasi pengalaman penggunaan permainan dan lebih umum hiburan interaktif. Bentuk evaluasi *User Experience* yang sangat mendasar adalah dengan hanya mencoba untuk memainkan permainan dan mencoba memahami bagaimana cara bermain tersebut. Pengenalan Video game jenis *Arcade* menunjukkan bahwa perubahan kecil dalam permainan game atau cerita sangat mempengaruhi pengalaman pengguna game secara keseluruhan (Novak, 2008). Saat ini berbagai metode dikerahkan untuk memahami berbagai hal yang dapat memberikan kontribusi dalam aspek keseluruhan bermain game (Federoff, 2002). Antarmuka atau kontroler *user* adalah peran penting dalam game. Evolusi kontroler game semakin berkembang dengan seiring berkembangnya jaman. Berinovasi bersama dengan grafik komputer Joystick analog menjadi peran penting untuk memberikan gerakan yang lebih akurat untuk menambah sensasi *User Experience* dalam memainkan game dengan sempurna. *User Experience* kontroler dengan menggunakan sensor adalah bukanlah hal baru dalam perkembangan kontroler game, dengan banyaknya penelitian seperti penggunaan sensor inersia dengan haptic untuk pengendalian game *Shooter* (Diaz, 2015).

Dalam pengendalian pada game Bus Simulator *user* akan memainkan dengan mode *first person view* yang mana dalam mode ini *user* mewajibkan

menggunakan *mouse* sebagai pengendalian pada *dashboard* game Bus Simulator dengan cara menggerakkan *mouse* ke kanan atau ke kiri. Dengan menggunakan sensor akselerometer memungkinkan bermain game Bus Simulator dengan *User Experience* yang berbeda.

Sensor Akselerometer adalah sebuah sensor yang umum digunakan untuk mengukur suatu percepatan terhadap objek. Teori yang digunakan pada sensor akselerometer adalah percepatan (akselerasi). Sensor Akselerometer mempunyai digital output yang dapat diakses melalui komunikasi I2C. Pada sensor ini dapat digunakan sebagai kendali suatu objek dengan contoh mengendalikan sebuah kursi roda dengan memanfaatkan sensor MPU9250 sebagai pengendali gerakan (Zakaria, 2017) dan sebagai alat penghitung langkah kaki yang memanfaatkan perhitungan langkah kaki dengan menggunakan sensor MPU9250 sebagai penghematan daya pada sistem (khalif, 2018).

Berdasarkan latar belakang diatas, ditarik penelitian untuk membuat perangkat *view controller*. Implementasi dilakukan dengan perangkat Arduino Pro Mikro dan sensor MPU9250 sebagai wujud dari *View Controller* melalui media perantara berupa Kabel serial dan video transfer *smartphone* dengan laptop untuk menambah *User Experience* (UX) dalam bermain game Bus Simulator.

1.2 Rumusan masalah

Berlandaskan latar belakang yang telah dijabarkan diatas, maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana merancang sistem *View Controller* menggunakan sensor Akselerometer pada game Bus Simulator sebagai sarana *User Experience* berbasis sistem *embedded*?
2. Bagaimana implementasi sistem *View Controller* menggunakan sensor Akselerometer pada game Bus Simulator sebagai sarana *User Experience* berbasis sistem *embedded*?
3. Bagaimana hasil survei dari sistem *View Controller* menggunakan sensor Akselerometer pada game Bus Simulator sebagai sarana *User Experience* berbasis sistem *embedded*?

1.3 Tujuan

Berlandaskan rumusan masalah yang telah dijabarkan diatas, maka dapat disimpulkan tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang sistem *View Controller* menggunakan sensor Akselerometer pada game Bus Simulator sebagai sarana *User Experience* berbasis sistem *embedded* untuk menambah *User Experience* (UX) dalam bermain game Bus Simulator.
2. Mendapatkan nilai akurasi perangkat Sensor Akselerometer MPU9250 yang dipasangkan di kepala (*Head band*) dari hasil implementasi sistem *View Controller* menggunakan sensor Akselerometer pada game Bus Simulator sebagai sarana *User Experience* berbasis sistem *embedded*.

3. Mendapatkan data *User Experience* (UX) pada sistem *View Controller* menggunakan sensor Akselerometer pada game Bus Simulator sebagai sarana *User Experience* berbasis sistem *embedded* oleh 30 user yang berbeda sebagai survei *User Experience* (UX).

1.4 Manfaat

Manfaat yang akan didapatkan dari penelitian rancang bangun perangkat *View controller*, yaitu:

1. Meningkatnya pengetahuan di bidang teknologi dan *gaming* dalam kehidupan sehari-hari.
2. Menciptakan pengalaman lebih dalam bermain game dengan memanfaatkan kemajuan teknologi.
3. Menjadi sarana *User Experience* (UX) dalam pemanfaatan pengembangan sebuah game.

1.5 Batasan masalah

Supaya penelitian ini bisa berjalan lancar, maka harus ada beberapa hal yang perlu dibatasi yaitu:

1. Perangkat menggunakan bantuan video transfer untuk memaksimalkan pengalaman bermain game Bus Simulator pada saat dipasang di kepala.
2. Bentuk fisik perangkat *View controller* dipasang di kepala.
3. Bentuk fisik perangkat adalah hasil produksi masal dan dapat dibeli secara legal.
4. Pada penelitian ini memiliki batasan permasalahannya yaitu berfokus menggantikan *mouse* sebagai *view controller* dengan menggunakan sensor MPU9250.

1.6 Sistematika pembahasan

Sistematika penulisan pada penelitian ini terdiri dari beberapa bab yang memberikan deskripsi dari penyusunan laporan penelitian yang dijelaskan sebagai berikut.

BAB I Pendahuluan

Pada bab ini berisi latar belakang penelitian, rumusan masalah penelitian, tujuan dari penelitian, manfaat dari penelitian, batasan masalah penelitian serta sistematika pembahasan.

BAB II Landasan Kepustakaan

Pada bab ini membahas kajian pustaka pada penelitian beserta dasar teori yang membantu dalam pembuatan sistem. Bab ini juga termuat penelitian-penelitian yang ada sebelumnya dan menjadi dasar pembuatan sistem.

BAB III Metodologi

Pada bab ini mengkaji tentang langkah-langkah dalam melakukan penelitian, antara lain studi literatur, pengumpulan data, analisa kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian, dan evaluasi sistem.

BAB IV Rekayasa Kebutuhan

Pada bab ini menerangkan secara keseluruhan kebutuhan sistem yang diperlukan, antara lain, kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan fungsional.

BAB V Perancangan dan Implementasi

Pada bab ini mengkaji tentang mekanisme perancangan dan implementasi sistem perangkat keras dan perangkat lunak untuk *view controller*.

BAB VI Pengujian dan Analisis

Pada bab ini membahas tentang cara pengujian, akurasi sensor, akurasi sistem, hasil akurasi sistem serta hasil korespondensi pengujian sistem.

BAB VII Kesimpulan

Pada bab ini akan membahas kesimpulan yang telah diperoleh dari perancangan, implementasi dan pengujian sistem serta saran-saran untuk pengembangan penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepastakaan mengandung landasan teori dan tinjauan pustaka yang diperlukan peneliti untuk penelitian ini serta mengkaji penelitian yang pernah ada kemudian diusulkan.

2.1 Tinjauan Pustaka

Memanfaatkan kursi roda sebagai bahan penelitian untuk membantu pengguna yang memiliki keterbatasan pada kaki dengan mengembangkan penelitian yang menggunakan pergerakan kepala sebagai kontroler berbasis *embedded* sistem untuk kendali pada kursi roda. Penelitian ini menggunakan sensor MPU6050 yang akan diletakkan pada kepala pengguna dengan 2 buah NodeMCU dimana 1 buahnya pada kontroler yang digunakan oleh *client* dan yang sisanya dipasang pada kursi roda sebagai server. Pada penelitian ini peneliti menggunakan complementary filter untuk masalah pembacaan nilai sudut yang disebabkan oleh noise dan dapat menghasilkan nilai yang lebih stabil dan akurat (Virza, 2017).

Dalam perkembangan jaman modern ini terjadi begitu pesat dalam dunia game. Game tinju adalah game *fighting* yang menggunakan teknik tusukan, silang dan bertahan sebagai tindakan *blocking* didalam game. Untuk melakukan beberapa tindakan tersebut, menggunakan Wii Remote sebagai kontroler dengan gerakan mengayun dimana gerakan tersebut menyerupai gerakan tinju asli. Tindakan yang dihasilkan akan dicocokkan menggunakan Euclidean Distance dengan pergerakan data masing-masing. Jika sudah memenuhi kriteria data *gesture* kemudian divisualisasikan dalam bentuk game dengan mode yang berbeda-beda dan dicocokkan dengan hasil persentase sukses atau tidaknya hasil dari tindakan yang dilakukan didalam game (Alfian, 2016).

Tabel 2.1 Penelitian Terkait

NO	Nama Penulis (Tahun), Judul Penelitian	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	(Virza Audy Ervanda ,2017), Pengembangan Sistem Deteksi Gerakan Kepala Sebagai Kontrol Pergerakan Kursi Roda Berbasis <i>Embedded Sistem</i>	Menggunakan Sensor MPU6050	Pengguna dapat melakukan kontrol kursi roda dengan memanfaatkan pergerakan kepala yang dirancang menggunakan sensor MPU6050.	Pengguna dapat memanfaatkan sensor MPU9250 sebagai <i>View controller</i> dengan memaksimalkan <i>User Experience</i> dalam bermain game.

Tabel 2.2 Penelitian Terkait (lanjutan)

NO	Nama Penulis (Tahun), Judul Penelitian	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
2	(Achmad Alfian Hidayat, dkk 2016), Development of <i>Fighting</i> Genre Game(Boxing) Using An Accelerometer Sensor	Menggunakan Sensor Akselerometer	Sistem dapat melakukan kontrol game dengan menggunakan sensor akselerometer	Pengguna dapat memanfaatkan sensor MPU9250 sebagai <i>View controller</i> dengan memaksimalkan <i>User Experience</i> dalam bermain game.

Dari beberapa penelitian diatas dapat disimpulkan bahwa memanfaatkan sensor akselerometer sebagai pengganti untuk beberpa kontrol adalah sangat memungkinkan. Dalam penelitian ini peneliti akan memanfaatkan sensor akselerometer sebagai pengganti pengendali *view controller* pada sebuah game Bus Simulator.

2.2 Dasar Teori

Pada dasar teori akan membahas tentang teori-teori yang akan diperlukan untuk menyusun penelitian yang telah diusulkan.

2.2.1 User Experience

User Experience (UX) adalah tanggapan dan persepsi seseorang yang dihasilkan dari penggunaan dan antisipasi sistem, produk atau layanan (ISO, 2010). Sederhananya, *User Experience* adalah bagaimana perasaan pengguna terhadap setiap interaksi yang sedang pengguna hadapi dengan apa yang ada didepan pengguna saat pengguna menggunakannya (Winter, 2015).

Untuk mendapatkan *User Experience* yang baik, maka sebuah sistem, produk atau layanan harus mempunyai 4 hal sebagai berikut:

1. Kesesuaian

Kesesuaian data *user experience* dengan fitur produk yang digunakan dengan kebutuhan pengguna.

2. Nilai

Menentukan produk tersebut berharga atau bernilai dengan kebutuhan pengguna.

3. Kemudahan

Menentukan kemudahan produk untuk pengguna atau melakukan hal-hal yang diinginkan oleh pengguna.

4. Perasaan

Menentukan perasaan pengguna dan dapat membuat perasaan pengguna senang saat menggunakannya.

Inilah empat elemen yang dibutuhkan untuk mendapatkan hasil *User Experience* yang baik (Guo, 2012).

2.2.2 Game Bus Simulator

Game Bus Simulator adalah game berjenis *Racing* dimana pengguna dapat merasakan sensasi bermain game dengan selayaknya seperti sedang menyetir Bus. Game ini sangat seru untuk dimainkan apalagi dengan menggunakan kontroler tertentu seperti kontroler kemudi atau pedal gas. Dalam game ini terdapat banyak misi yang dijalankan termasuk mengantarkan penumpang dengan tujuan tertentu (bussimulator-game.com, 2018).



Gambar 2.1 Game Bus Simulator

Sumber: steampowered.com (2018)

Game ini menawarkan 2 macam *view* sudut pandang yaitu *long view* dan *first person view*. Dalam mode *first person view* pengguna berada didalam *dashboard* bus sehingga pengguna dapat merasakan *User Experience* (UX) selayaknya berada di dalam *dashboard* bus sesungguhnya seperti dalam Gambar 2.2 dibawah ini :



Gambar 2.2 First person mode

Sumber: steampowered.com (2018)

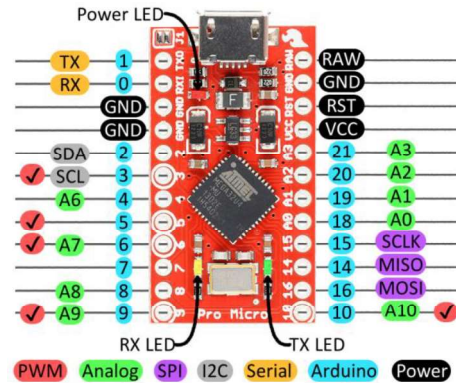
Pada Gambar 2.4, pengendalian dalam game Bus Simulator *user* akan memainkan dengan mode *first person view* yang mana dalam mode ini *user* mewajibkan menggunakan *mouse* sebagai pengendalian pada *dashboard* game Bus Simulator dengan cara menggerakkan *mouse* ke kanan atau ke kiri. Pada game Bus Simulator ini selain mengemudi *user* juga harus menjaga ketertiban di dalam bus, menjual tiket dan memasukkan penumpang dapat *user* rasakan seakan seperti mengemudikan bus secara nyata. Pada game ini juga dapat menampilkan rute-rute yang ingin dituju. Dengan mematuhi rambu-rambu lalu lintas yang ada dan memanfaatkan kamera mode *first person view* akan menambah pengalaman berkendara *user*. Berikut rekomendasi sistem dalam game Bus Simulator :

Tabel 2.2 Rekomendasi Sistem Game Bus Simulator

No	Nama	Keterangan
1	OS	Windows 7/8/10 64 bit
2	Prosesor	Intel Core i5 with 3.40 GHz
3	Memori	8 GB RAM
4	Grafik	GeForce GTX 660 (2 GB VRAM)
5	DirectX	Versi 9.0
6	Penyimpanan	5 GB

2.2.3 Arduino Pro Mikro

Arduino Pro Mikro adalah mikrokontroler yang berdasarkan ATmega32u4. Pada Gambar 2.2, Arduino Pro Mikro memiliki 20 pin input atau output digital, dijelaskan Pada Gambar 2.2 dimana 7 pin dapat digunakan sebagai output PWM dan 12 pin sebagai input analog, osilator 16 MHz, koneksi USB mikro, header ICSP, dan tombol reset. Arduino Mikro memiliki komunikasi USB *built-in* yang mana dapat menghilangkan kebutuhan prosesor sekunder.



Gambar 2.3 Diagram komponen Arduino Pro Mikro

Sumber: sparkfun.com (2018)

Pada Gambar 2.2, daya yang dibutuhkan Arduino Pro Mikro dapat melalui koneksi *micro* USB atau dengan catu daya eksternal. Arduino Pro Mikro dapat beroperasi dengan suplai daya eksternal dari 6 sampai 20 volt. Jika dipasang dengan kurang dari 7 volt, *board* tidak akan stabil.

Arduino Pro Mikro dapat berkomunikasi dengan sejumlah fasilitas dari Arduino lainnya atau mikrokontroler yang lain. ATmega32U4 menyediakan komunikasi serial UART (*Universal Asynchronous Reciever transmitter*) yang tersedia pada pin digital 0 (RX) dan 1 (TX). ATmega32U4 juga bisa digunakan pada komunikasi serial CDC (*Communication Device Class*) melalui USB. Karakteristik Arduino Pro Mikro seperti pada table dibawah ini :

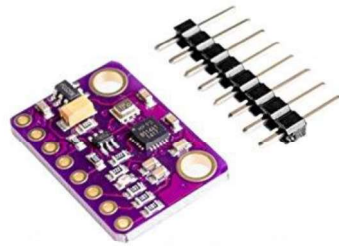
Tabel 2.3 Karakteristik Arduino pro mikro

No	Parameter	Nilai
1	Operasi Tegangan	5 V
2	Mikrokontroler	ATMega32u4
3	SRAM	2,5 KB
4	EEPROM	1 KB
5	Arus DC setiap I/O pin	40mA

Sumber: farnell.com (2018)

2.2.4 MPU9250 GY-91

MPU9250 adalah modul multi chip (MCM) yang memiliki 10 sumbu *Motion Tracking* perangkat yang menggabungkan 3 sumbu giroskop, 3 sumbu akselerometer, 3 sumbu magnetometer dan barometric sensor serta *Digital Motion Processor* (DMP).



Gambar 2.4 MPU9250 GY-91

Sumber: playground.arduino.cc (2018)

MPU9250 adalah generasi kedua dari perangkat 9-axis *Motion Tracking* untuk *smartphone*, tablet, dan wearable sensor. Bentuk desain dari MPU9250 sangatlah kecil dengan ukuran 3x3x1mm yang mana menjadikan MPU9250 model GY-91 sebagai perangkat terkecil dari kelas 9 axis *Motion Tracking* serta menggabungkan inovasi desain *InvenSense* terbaru yang mana dapat mengurangi konsumsi daya dan meningkatkan kinerja.

Pada Gambar 2.4, terdapat beberapa pin yang ada pada sensor MPU9250 yang akan digunakan pada penelitian ini, berikut beberapa kegunaannya:

1. Voltage Supply Pin.
2. 3.3v regulator output.
3. 0 power supply (GND).
4. I2C clock/SPI clock (SCL).
5. I2C data / SPI data input (SDA).
6. SPI data output / I2C slave address configuration pin (SD0/SA0).
7. Chip select for SPI mode only for MPU9250 (NCS).
8. Chip select for BMP280 (CSB).

Untuk karakteristik dari MPU9250 adalah sebagai berikut :

Tabel 2.4 Karakteristik MPU9250

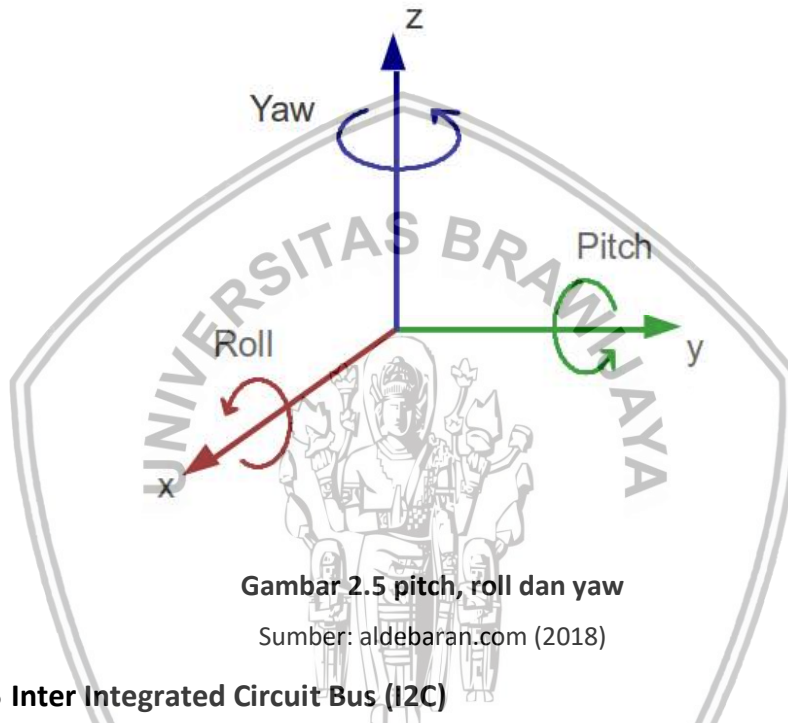
No	Parameter	Keterangan
1	<i>Model</i>	GY-91
2	<i>Sensor chips</i>	MPU9250 + BMP280
3	<i>Operating voltage range</i>	3.0V 5.0V
4	<i>Gyroscope range</i>	250, 500, 1000/s

Tabel 2.4 Karakteristik MPU9250 (Lanjutan)

No	Parameter	Keterangan
5	Acceleration range	2 4 8 16g

Sumber: store.invensense.com (2018)

Pada sensor MPU9250 terdapat sumbu *pitch*, *roll* dan *yaw*. *Pitch* merupakan gerakan mengangguk atau gerakan keatas dan kebawah, *rolling* merupakan gerakan berguling, dan *yawing* merupakan gerakan menggeleng ke kanan dan ke kiri (www.grc.nasa.gov, 2014).

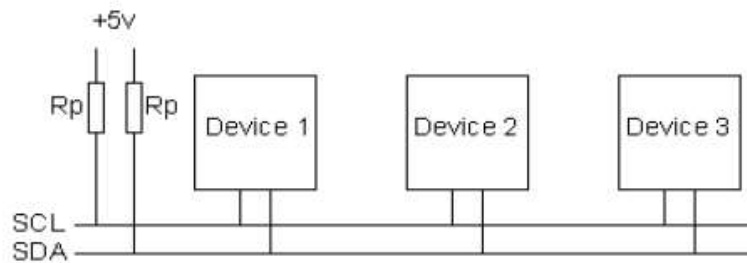


Gambar 2.5 pitch, roll dan yaw

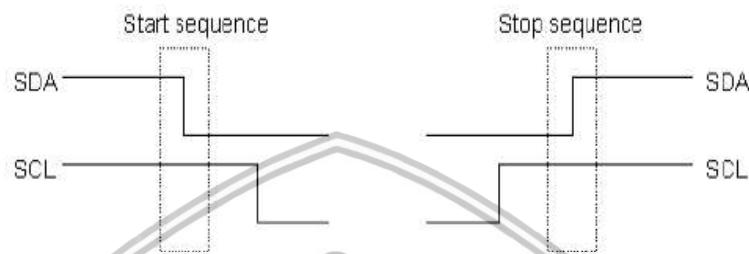
Sumber: aldebaran.com (2018)

2.2.5 Inter Integrated Circuit Bus (I2C)

Protokol I2C adalah protocol serial yang umum digunakan karena kemudahan dan efektivitas biaya yang digunakan. Bus I2C terdiri dari 2 baris: SDA, SCL. SDA adalah jalur data yang membantu dalam membawa data dari satu perangkat ke perangkat lain dan SCL adalah *clock line* yang membantu dalam sinkronisasi semua perangkat yang terhubung menggunakan bus I2C. representasi diagram blok dari I2C bus seperti ditunjukkan pada Gambar 2.5 di bawah ini :



(a)



(b)

Gambar 2.6 (a) Diagram blok I2C (b) Start dan stop sequence I2C

Perangkat yang terhubung ke bus I2C juga bisa dikonfigurasi sebagai master atau slave. Node master umumnya yang menghasilkan sinyal clock.

2.2.6 Arduino IDE

IDE adalah merupakan singkatan dari *Integrated Development Environment*, atau menurut bahasa mudahnya adalah tempat integrasi yang digunakan untuk pengembangan. Disebut sebagai tempat karena dengan menggunakan *software* ini sehingga Arduino dapat dilakukan pemrograman untuk melakukan fungsi-fungsi yang ditanamkan menggunakan sintaks pemrograman. Arduino ini menggunakan bahasa pemrograman C.

Arduino IDE dibuat dari Bahasa pemrograman JAVA. Arduino IDE dilengkapi dengan beberapa *library* C/C++ yang biasa disebut *wiring* yang bisa membuat operasi input dan output menjadi lebih mudah.

2.2.7 Library Mouse.h

Adalah fungsi *library* yang ada pada Arduino IDE untuk komunikasi serial I2C pada komputer yang bertujuan untuk menggantikan *mouse* dengan MPU9250. Fungsi *mouse* memungkinkan Arduino berbasis mikro 32u4 untuk mengontrol pergerakan kursor pada komputer yang terhubung melalui port USB dari komputer (Arduino.cc, 2014).

2.2.8 Library IMU MPU6050.h

Library MPU6050.h merupakan fungsi *library* didalam Arduino IDE yang dapat digunakan untuk meringankan dalam menggunakan sensor dengan IMU

MPU6050, fungsi-fungsi didalamnya yang ada seperti pembacaan data *raw* serta pembacaan sudut *roll*, *pitch* dan *yaw*. (Arduino.cc, 2014).

2.2.9 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) merupakan sebuah bentuk standar deviasi residual (kesalahan prediksi). Residual merupakan sebuah ukuran seberapa jauh dari titik data sebuah garis regresi. Untuk mengetahui parameter penyebaran residual adalah dengan menggunakan RMSE. Secara umum rumus dari RMSE adalah:

$$RMSE = \sqrt{\frac{1}{2} \sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}} \quad (2.1)$$

\hat{y}_i merupakan nilai dari prediksi dan y_i merupakan nilai dari hasil yang diobservasi. Persamaan RMSE bisa menghitung *error* deviasi dengan baik tetapi tidak bisa digunakan untuk membandingkan hasil *error* antar variabel dalam suatu sistem. (Latief Nurrohman, 2018).

2.2.10 Skala Likert

Skala Likert merupakan tipe skala dengan menggunakan perantara angket atau kuisisioner dalam sebuah survei. Nilai skala dari pernyataan akan ditentukan oleh distribusi respon dari responden. Persamaan dari skala likert adalah:

$$\text{Rumus Index \%} = \frac{\text{Total Skor}}{Y \times 100} \quad (2.2)$$

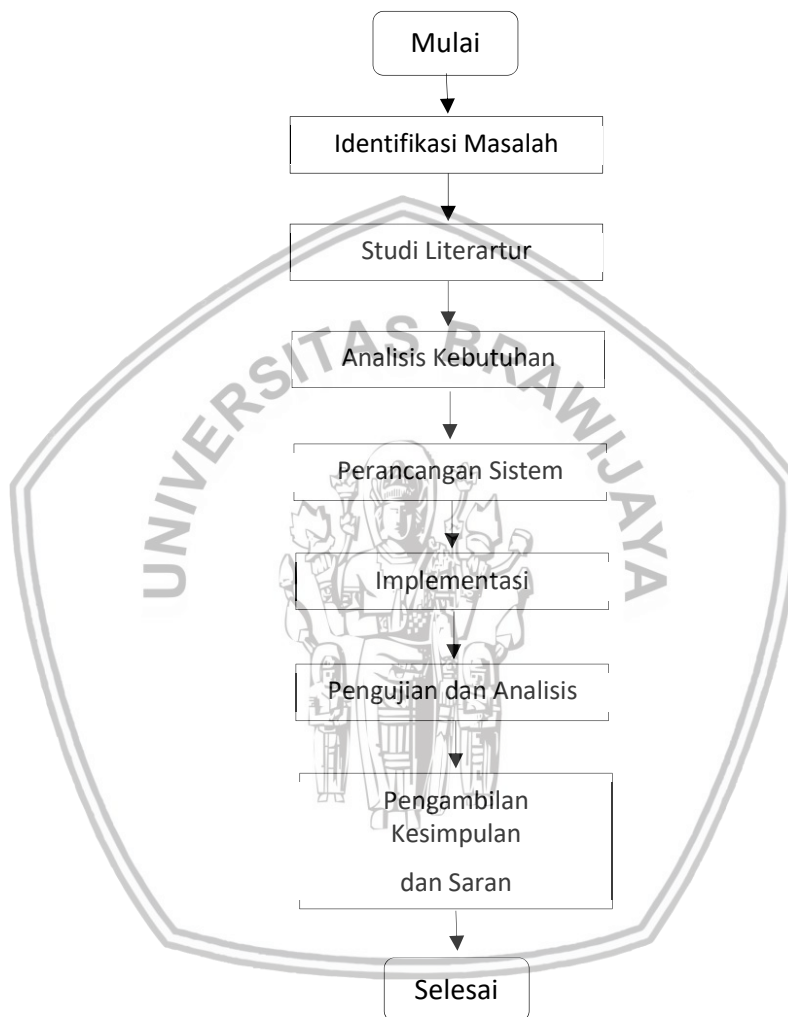
Total skor merupakan kumpulan nilai yang didapat dari beberapa pertanyaan yang dijawab oleh *user* dan Y merupakan parameter skor paling tinggi yang didapatkan dari sebuah pertanyaan. Untuk dapat mengetahui nilai dari skala likert maka digunakan persamaan sebagai berikut:

$$I = \frac{100}{\text{Jumlah Skor}} \quad (2.3)$$

Dimana I merupakan selang nilai yang dicari untuk dapat mengetahui penilaian sedangkan jumlah skor merupakan jumlah dari total skor yang telah disediakan. (Latief Nurrohman, 2018).

BAB 3 METODOLOGI

Dalam penelitian ini terdapat beberapa proses yang saling berkaitan satu sama lain agar perancangan penelitian ini terstruktur dengan rapi dan baik. Studi dalam penelitian ini adalah rancang bangun sistem *View controller*. Berikut alur metodolgi dalam penelitian ini:



Gambar 3.1 Alur Metodologi

Pada Gambar 3.1, menunjukan urutan diagram alir dalam melakukan penelitian. Dimulai dengan Analisis kebutuhan, dimana pada tahap ini dilakukan tahap pengumpulan suatu data terhadap parameter yang digunakan dalam penelitian ini. Perancangan sistem, dimana pada tahap ini membahas tentang pemodelan sistem. Implementasi, pada tahap ini dilakukan proses integrasi antara perangkat keras dan perangkat lunak. Pengujian dan analisis, pada tahap ini akan dilakukan pengujian sistem serta menganalisis. Pengambilan kesimpulan dan

saran, pada tahap ini dilakukan penarikan kesimpulan serta saran guna untuk mengembangkan sistem lebih lanjut.

3.1 Identifikasi Masalah

Tahap identifikasi masalah adalah bertujuan untuk memfokuskan tujuan penelitian ini terhadap rumusan masalah yang sedang diteliti. Dalam identifikasi masalah terdapat latar belakang penelitian, tujuan penelitian, manfaat penelitian dan batasan-batasan dalam penelitian.

3.2 Studi Literatur

Studi Literatur dilakukan untuk mengumpulkan tinjauan pustaka dan dasar teori yang dapat mendukung untuk pengerjaan penulisan laporan penelitian. Studi literatur dikerjakan dengan cara mengumpulkan beberapa teori dan pustaka yang berkaitan dengan penelitian yang meliputi sebagai berikut:

1. Sensor MPU9250
2. *Inter Integrated Circuit Bus (I2C)*
3. Mikrokontroler Arduino Pro Mikro
4. Pemrograman Arduino
5. *User Experience (UX)*
6. Game Bus Simulator
7. *Library mouse.h*
8. *Library IMU MPU6050.h*
9. Skala Likert
10. *Root Mean Square Error (RMSE)*

3.3 Analisis Kebutuhan

Dalam penelitian ini dilakukan analisis untuk mendapatkan suatu kebutuhan dalam penelitian. Analisis kebutuhan yang dibutuhkan yaitu kebutuhan berupa komponen-komponen perangkat keras serta kebutuhan pada piranti lunak. Analisis kebutuhan sistem ini akan digunakan sebagai petunjuk dalam penelitian.

3.3.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras dalam penelitian ini seperti :

1. Mikrokontroler Arduino Pro Mikro
2. MPU9250
3. Kabel *Jumper*
4. *Headband VR google card box*
5. Laptop
6. *Smartphone*

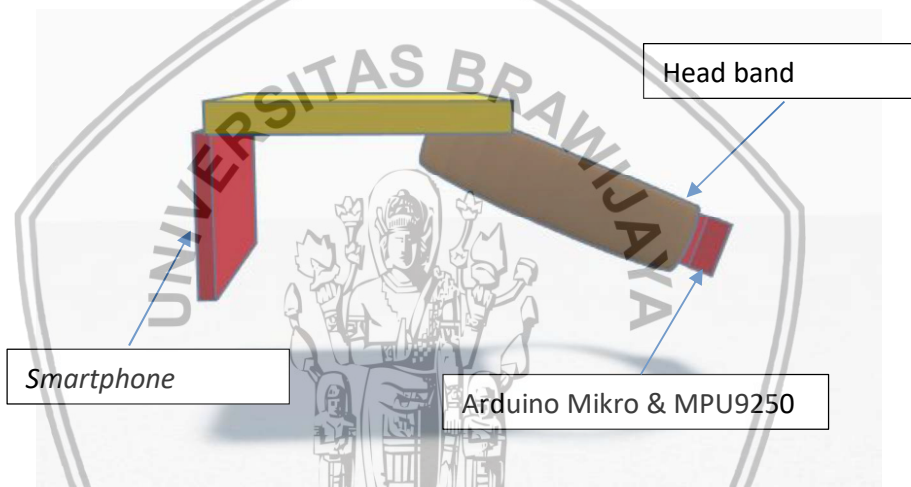
3.3.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat keras dalam penelitian ini seperti :

1. Arduino IDE
2. *Library Mouse.h*
3. *Library IMU MPU6050.h*
4. Aplikasi Trinus VR
5. Game Bus Simulator

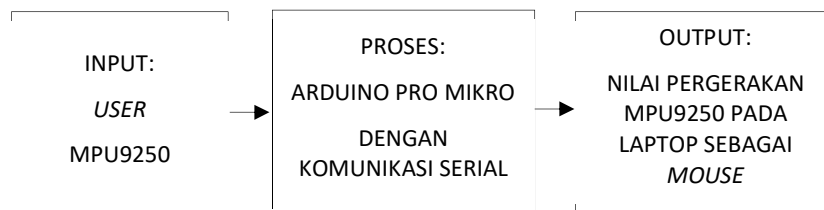
3.4 Perancangan Sistem

Pada perancangan sistem dibuat diagram blok yang memiliki bagian-bagian dari beberapa sistem yang dibuat. Bagian-bagian tersebut mempunyai tugas sesuai dengan tujuan penelitian, kemudian diimplementasikan pada sistem rancang bangun *View controller*. Adapun rancangan *View controller* yang akan dibuat sebagai berikut:



Gambar 3.2 Perancangan *View Controller* menggunakan Sensor MPU9250

Pada Gambar 3.2, Sensor Akselerometer MPU9250 merupakan perangkat keras yang digunakan untuk membaca gerakan yang diperoleh dari kepala, MPU9250 diletakkan di kepala. Fungsi dari MPU9250 adalah untuk identifikasi setiap gerakan dari *user* yang memakai *device View controller*.



Gambar 3.3 Diagram blok perangkat keras

Pada Gambar 3.3, Arduino pro mikro merupakan sebuah sistem kendali perangkat keras yang digunakan peneliti. MPU9250 akan digunakan untuk membaca nilai yang ada pada kepala *user*, lalu nilai tersebut diolah dan mengirimkan data yang telah dibaca dari MPU9250. Pada tahap selanjutnya Arduino Mikro melakukan komunikasi dengan cara serial untuk mengirimkan data MPU9250 yang telah dibaca ke laptop. Laptop berfungsi untuk memainkan game Bus Simulator.

3.5 Implementasi

Implementasi sistem ini mengacu pada perancangan sistem yang telah dibuat sebelumnya. Pada implementasi ini mencantumkan beberapa gambar dari sistem yang dibuat oleh peneliti serta potongan-potongan bahasa pemrograman yang digunakan oleh peneliti.

3.6 Pengujian dan Analisis

Pengujian pada penelitian ini menggunakan metode skenario pengujian. Adapun beberapa skenario :

3.6.1 Pengujian nilai sudut perangkat *view controller*

Prasyarat pengujian perangkat *view controller* meliputi fungsional perangkat keras dan keterikatan piranti lunak didalamnya. Pengujian ini adalah pengujian terhadap perangkat keras dan perangkat lunak didalamnya apakah nilai sudut yang didapatkan dari sensor mendapatkan nilai RMSE yang rendah sehingga nilai sudut tersebut sudah siap diimplementasikan.

3.6.2 Pengujian fungsional dari *view controller*

Pengujian menggunakan verifikasi nilai yaw dari MPU9250 dengan satu kondisi. Skenario yang dimaksudkan menggunakan game Bus Simulator sebagai penerapan *View controller* dengan penerapan kondisi kepala menoleh ke kanan dan ke kiri sebagai batasan nilai yang di dapatkan yaitu 40 derajat.

3.6.3 Pengujian *user* yang akan di uji coba dalam penggunaan *view controller* dengan korespondensi kuisisioner

Pengujian ini menggunakan metode survei deskriptif kuantitatif dengan melibatkan 30 *user* untuk menghasilkan data berbeda dalam penggunaan *View controller* pada game Bus Simulator dengan mencoba perangkat *view controller* kemudian mengisi kuisisioner yang telah disediakan sebagai bahan survei *user experience* dari penggunaan perangkat *view controller*.

3.7 Penarikan Kesimpulan

Penarikan kesimpulan dilakukan setelah semua tahap demi tahap telah dilakukan oleh peneliti. Kesimpulan yang dihasilkan harus bisa menjawab rumusan masalah yang ada pada Bab 1. Bagian akhir dari penelitian ini berisi saran dengan tujuan untuk penyempurnaan penelitian selanjutnya.

BAB 4 REKAYASA KEBUTUHAN

Pada bab ini akan menjelaskan secara rinci gambaran umum sistem, kebutuhan fungsional, kebutuhan perangkat keras, kebutuhan perangkat lunak dan batasan sistem.

4.1 Gambaran Umum Sistem

Sistem yang akan dibuat yaitu *view controller* pada game Bus Simulator dengan menggunakan sensor MPU9250. *View Controller* ini terdiri dari mikrokontroller Arduino Pro Mikro dan Sensor MPU9250. Cara kerja dari *view controller* ini yaitu pada sensor MPU9250 akan mendeteksi gerakan yang dilakukan oleh *user* lalu data hasil dari *sensing* akan diolah oleh Arduino Pro Mikro kemudian dikirimkan komputer. Komunikasi yang digunakan adalah dengan menggunakan komunikasi I2C yang komunikasinya 2 arah secara serial. Komunikasi serial I2C pada Arduino memungkinkan untuk melakukan pengiriman data dari master writer (Arduino) menuju slave receiver (komputer). Pada penelitian ini memiliki batasan permasalahannya yaitu berfokus menggantikan *mouse* sebagai *view controller* dengan menggunakan sensor MPU9250. Sistem menggunakan *smartphone* sebagai perantara stream dari laptop dengan menggunakan jaringan akses *point* yang sama. Dengan menggunakan *streaming* akan memudahkan *user experience* (UX) terhadap game bus simulator dengan sistem yang digunakan.

4.2 Karakteristik Pengguna

Karakteristik pengguna dilakukan untuk mengetahui karakteristik yang dibutuhkan oleh pengguna pada sistem yang dibuat, diantaranya adalah sebagai berikut :

1. Umur *user*

Merupakan kebutuhan karakteristik pengguna terhadap sistem yang dibuat dengan menentukan umur pengguna yang dibatasi dari umur 20-25 tahun. Dikarenakan penyesuaian terhadap ukuran kepala yang akan diimplementasikan terhadap sistem.

2. Kebiasaan *user*

Merupakan kebutuhan karakteristik yang ditunjukkan kepada pengguna yang tidak memakai kacamata dikarenakan penyesuaian terhadap perancangan sistem yang dibuat.

3. Cara bermain *user*

Merupakan kebutuhan karakteristik pengguna untuk memainkan game bus simulator dengan cara duduk tegap dengan menggerakkan kepala ke kanan dan ke kiri.

4.3 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mengetahui kebutuhan yang dibutuhkan oleh sistem. Kebutuhan yang dibutuhkan adalah kebutuhan fungsional dan non fungsional. Kebutuhan perangkat keras dan kebutuhan perangkat lunak termasuk dalam kebutuhan non fungsional sistem.

4.3.1 Kebutuhan Fungsional

Kebutuhan fungsional yang harus bisa dijalankan oleh sistem adalah :

1. Sensor MPU9250 dapat mengakuisisi nilai posisi dan orientasi kepala.

Merupakan kebutuhan sistem yang terkait dengan kendali *view controller*. Pada kebutuhan sistem ini sensor MPU9250 akan mendeteksi nilai posisi yang telah didapatkan melalui hasil percobaan yang dilakukan oleh *user*. Nilai yang didapat akan menjadikan masukan yang akan dibutuhkan untuk mengendalikan *view controller* pada game Bus Simulator.

2. Arduino Pro Mikro dapat menerima data dari sensor MPU9250 dan mengirimkan data ke komputer.

Arduino Pro Mikro dapat menerima nilai data yang dihasilkan oleh sensor MPU9250 dan mengolahnya lalu dikirimkan ke komputer. Dengan hasil *sensing* yg telah dilakukan oleh sensor MPU9250 dapat digunakan sebagai *view controller* pada game Bus Simulator.

3. Hasil *sensing* dari sensor MPU9250 dapat menggantikan *mouse* sebagai *view controller* pada game Bus Simulator.

Fungsionalitas ini berfungsi untuk menggantikan *mouse* pada komputer sebagai *view controller* pada game Bus Simulator dengan memanfaatkan hasil *sensing* yang dilakukan oleh sensor MPU9250 terhadap *user* yang memainkan game Bus Simulator.

4. Sistem dapat mengenali gestur *user* menoleh ke kanan dan ke kiri.

Fungsionalitas ini berfungsi untuk mengetahui gerakan dari *user* dengan mengenali gestur menoleh ke kiri dan ke kanan pada sistem sebagai cara melihat *view* didalam *dashboard* game Bus Simulator.

4.3.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional dari sistem ini terdiri dari kebutuhan perangkat keras dan perangkat lunak yang digunakan oleh sistem.

4.3.2.1 Kebutuhan Perangkat keras

Pada bagian ini menjelaskan terkait dengan kebutuhan perangkat keras pada sistem ini. Perangkat keras ini meliputi dari sekumpulan komponen elektronika yang akan membentuk suatu sistem perangkat keras yang memiliki fungsi tertentu. Berikut adalah kebutuhan perangkat keras yang digunakan, yakni sebagai berikut :

1. Mikrokontroler Arduino Pro Mikro

Mikrokontroler ini digunakan untuk mengolah sinyal dan sebagai perangkat keras yang dapat mengolah masukan dari sensor, atau dapat disebut sebagai otak dari sebuah sistem.

2. MPU9250

Sensor MPU9250 ini digunakan untuk mendapatkan nilai dari gestur gerakan yang dilakukan oleh *user* dengan menoleh ke kanan dan ke kiri yang mana dari hasil gerakan tersebut nilainya akan dikirim ke mikrokontroler sebagai hasil sensing gestur untuk menggerakkan *view controller* pada game Bus Simulator.

3. Kabel jumper

Kabel jumper berguna untuk menghubungkan beberapa perangkat dan sensor untuk saling terhubung dan bekerja.

4. Headband VR card box

Headband VR card box digunakan sebagai perantara *view controller* dengan memasang komponen-komponen dari sistem ke headband VR google card box.

5. Laptop

Berguna untuk memprogram mikrokontroler dan sebagai penyedia daya bagi mikrokontroler.

6. *Smartphone*

Berguna untuk visualisasi game dari laptop ke *smartphone*.

4.3.2.2 Kebutuhan Perangkat lunak

Berikut adalah kebutuhan perangkat lunak yang digunakan pada proses perancangan sistem, yakni sebagai berikut :

1. Arduino IDE

Arduino IDE merupakan *Software IDE (Integrated Development Environment)* atau secara bahasa merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan program. *Software* inilah yang menjadi pen jembatan antara sintaks-sintaks pemrograman yang akan diimplementasikan ke dalam *board* Arduino itu sendiri. Bahasa pemrograman yang digunakan adalah bahasa pemrograman berbasis JAVA

yang dilengkapi dengan *library* C/C++ yang biasa disebut *Wiring* yang dapat membantu operasi masukan dan keluaran menjadi lebih mudah.

2. *Library* Mouse.h

Merupakan *library* yang digunakan untuk komunikasi serial I2C pada komputer yang bertujuan untuk menggantikan *mouse* dengan MPU9250.

3. *Library* IMU MPU6050.h

Merupakan *library* yang digunakan untuk mendapatkan data *digital motion processing* (DMP) dari sensor MPU9250.

4. Aplikasi Trinus VR

Merupakan software yang digunakan untuk mentransfer video dari laptop ke *smartphone* sebagai bentuk *User Experience* untuk memainkan game.

5. Game Bus Simulator

Merupakan game yang digunakan untuk menguji perangkat *view controller*.

4.4 Batasan Desain Sistem

Pembuatan *view controller* pada game Bus Simulator ini mempunyai beberapa batasan. Adapun batasan desain sistem ini adalah :

1. Sistem melakukan pendeteksian dengan memainkan game Bus Simulator dengan menggelengkan kepala ke kanan dan ke kiri sebagai uji dalam sistem.
2. Sistem hanya mendeteksi nilai yang didapat oleh sensor MPU9250.
3. Sistem menggunakan bantuan *headband* dari perangkat VR yang telah dimodifikasi agar sesuai dengan rancangan sistem *view controller*.
4. Sistem dibantu dengan menggunakan perangkat keras *smartphone* untuk mentransfer video game dari laptop agar penggunaan sistem dapat maksimal.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini memberikan penjelasan mengenai perancangan dan proses implementasi sistem yang meliputi perancangan sistem menggunakan diagram blok perangkat keras, skematik perangkat keras kemudian beberapa diagram alir yang ditanam pada perangkat. Implementasi sistem terdiri dari implementasi perangkat keras serta perangkat lunak.

5.1 Perancangan Sistem

Pada bagian ini dijelaskan proses perencnaan sistem yang dimulai dari perancangan perangkat keras meliputi diagram blok serta skematik perangkat keras dan perancangan perangkat lunak berupa diagram alir yang dijelaskan menggunakan *flow chart*.

5.1.1 Diagram Blok Sistem

Pada sistem *view Controller* menggunakan sensor Akselerometer pada game Bus Simulator sebagai sarana *user experience* berbasis sistem *embedded* ini menerapkan 2 perangkat keras yang meliputi perangkat keras sensor MPU9250, dan Arduino Pro Mikro. Kedua perangkat tersebut mempunyai fungsionalitas yang berbeda, perangkat keras Sensor MPU9250 mempunyai fungsionalitas untuk membaca nilai gerak dari gerakan kepala menoleh ke kanan dan ke kiri. Sedangkan perangkat keras Arduino Pro Mikro mempunyai fungsionalitas untuk mengolah data hasil dari *sensing* yang dilakukan oleh sensor MPU9250 untuk diolah menjadi nilai yang dapat menggerakkan kursor *mouse* pada laptop sebagai *view controller* pada game Bus Simulator.

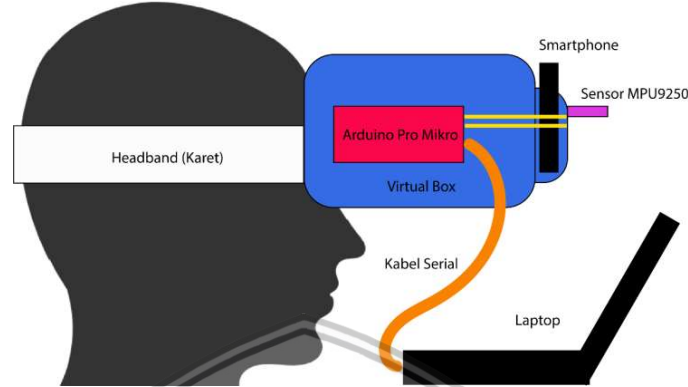


Gambar 5.1 Diagram blok sistem *View Controller*

Pada Gambar 5.1, Mikrokontroler yang digunakan pada sistem adalah menggunakan mikrokontroler berbasis ATmega32U4. Peran yang dimiliki perangkat keras sensor MPU9250 adalah dapat membaca nilai *yaw*, *pitch*, dan *roll* dari hasil *sensing* yang telah dilakukan oleh *user* untuk kemudian di olah menjadi *kursor* pada laptop oleh mikrokontroler Arduino pro mikro. Sistem hanya menggunakan sensor MPU9250 sebagai input, karena pada perancangan ini sistem hanya memerlukan input masukan berupa gerakan menoleh ke kanan dan ke kiri.

5.1.2 Perancangan alat dan peletakan *view controller*

Dalam perancangan alat dan peletakan *view controller* menggunakan sensor MPU9250, mikrokontroler arduino pro mikro yang diletakkan pada *virtual box* yang telah pasang *headband* karet untuk penggunaan di kepala.

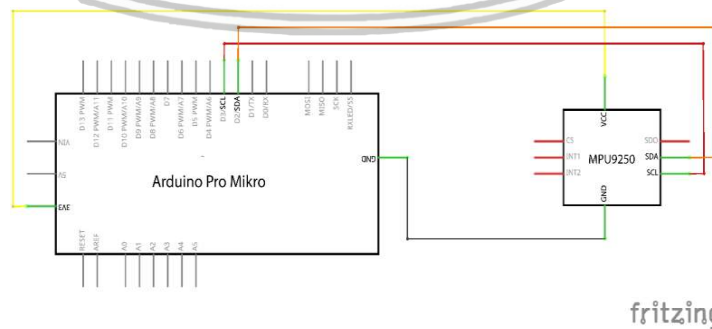


Gambar 5.2 Ilustrasi perancangan alat tampak samping pada sistem *view controller*

Pada Gambar 5.2 dijelaskan bahwa pada *virtual box* terdapat sensor MPU9250, *smartphone* dan mikrokontroler arduino pro mikro yang terpasang pada posisi yang telah disesuaikan sehingga mendapatkan gambaran perancangan perangkat keras seperti pada gambar. Komunikasi yang digunakan oleh sistem *view controller* adalah dengan menggunakan kabel serial, selain sebagai komunikasi serial dengan laptop, kabel komunikasi serial juga berfungsi sebagai penghubung power dari perangkat *view controller*. Dijelaskan juga bahwa laptop adalah sebagai power dari *view controller*. Selain menjadi power, laptop juga akan menjalankan game bus simulator yang akan digunakan oleh sistem *view controller*.

5.2 Perancangan Perangkat Keras

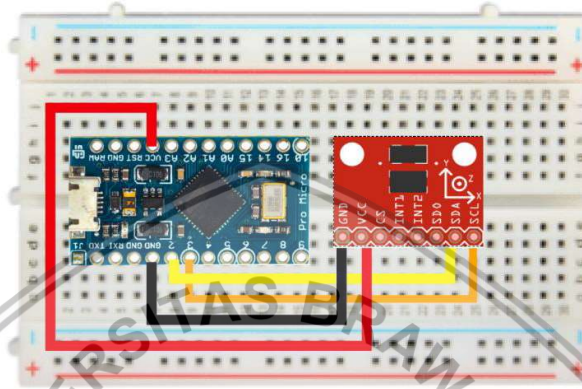
Tahap perancangan perangkat keras berupa skematik ini penulis menggunakan perangkat lunak Fritzing dan Adobe Illustrator. Skematik memiliki fungsi sebagai pendekatan pada seluruh perangkat keras terkait hubungan antara sensor, mikrokontroler, serta media komunikasi.



Gambar 5.3 Skema perencanaan perangkat keras sistem *view controller*

Pada blok diagram diatas, input data adalah berupa gerakan. Selanjutnya data pada sensor MPU9250 yang telah diambil akan di proses pada Arduino sehingga sistem mengetahui gerakan yang dihasilkan. Kemudian hasil dari proses yang dilakukan pada Arduino akan diolah menjadi *mouse* pada laptop.

Untuk komunikasi antara Arduino Pro Mikro dan MPU9250 menggunakan I2C. MPU9250 digunakan sebagai *master writer* dan Arduino Pro Mikro sebagai *slave receiver*. Pada Gambar 5.4, merupakan gambar dari koneksi I2C antara Arduino Pro mikro dan MPU9250.



Gambar 5.4 Konfigurasi sambungan SCL dan SDA

Pada Gambar 5.4, digambarkan bahwa sensor akselerometer MPU9250 terdiri dari 8 pin. Untuk terhubungnya antara MPU9250 dengan mikrokontroler membutuhkan 4 pin dari masing-masing perangkat yakni 1 pin yang dibutuhkan sebagai masukan tegangan yang dihubungkan pada pin 3v3 yang ada pada mikrokontroler, 1 pin terhubung dengan *ground* pada mikrokontroler, 1 pin untuk SDA dari MPU9250 terhubung pada pin 2 pada mikrokontroler dan 1 pin SCL dari MPU9250 terhubung pada pin 3 pada mikrokontroler. Untuk fungsionalitas dari masing-masing pin dari perangkat *view controller* sudah dijelaskan pada dasar teori pada bab sebelumnya.

Koneksi pin pada Gambar 5.3 dan 5.4 ditunjukkan pada Tabel 5.1 berikut

Tabel 5.1 Koneksi pin perancangan perangkat keras sistem

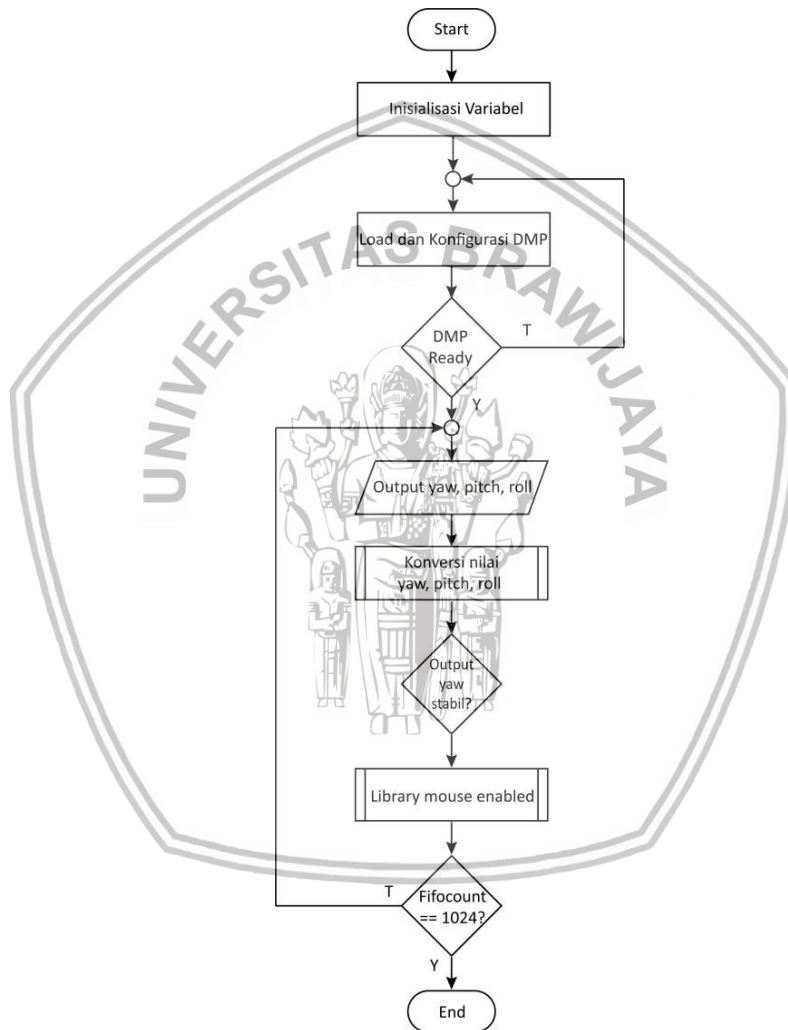
No	Pin MPU9250	Pin Arduino Pro Mikro
1	3v3	Vcc
2	GND	GND
3	SDA	2
4	SCL	3

5.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak dibahas menjadi tiga bagian. Pertama perancangan program utama. Kedua perancangan konversi *pitch* dan *yaw*. Dan yang ketiga perancangan *library mouse* pada Arduino.

5.3.1 Perancangan kode program utama

Pada perancangan kode program utama ini mencakup semua proses dari sistem. Dari mendapatkan nilai sensor sampai menggerakkan *mouse* untuk game bus simulator. Perancangan kode program utama akan dijelaskan dengan diagram alir seperti dibawah ini.



Gambar 5.5 Diagram alir kode program utama

Pada diagram alir Gambar 5.5, proses sistem yang ada pada *view controller* digambarkan dengan menggunakan *flowchart* sehingga dapat memberikan gambaran umum sistem pada sisi perangkat lunak dalam *view controller*. Pada *view controller* terjadi beberapa proses utama yaitu mengaktifkan *Digital Motion*

Processing (DMP) dari sensor MPU9250 dan mengaplikasikan *library mouse* sebagai output dari sistem *view controller*.

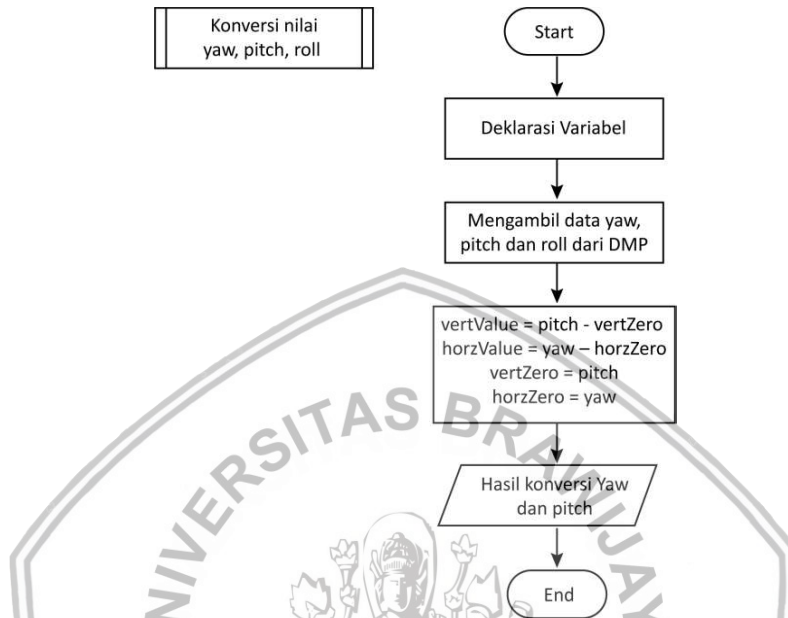
Proses mengaktifkan *Digital Motion Processing* (DMP) adalah bertujuan untuk mendapatkan nilai dari hasil *sensing*. Setelah mendapatkan data dengan nilai *pitch*, *roll*, dan *yaw* kemudian data tersebut diolah kedalam *library mouse* untuk dijadikan output berupa *cursor* pada laptop yang bertujuan menjadi *view controller* pada game Bus Simulator.

Pada Gambar 5.5, dijelaskan mengenai beberapa tahapan diagram alir kode program utama dari mulai hingga selesai. Berikut penjelasan dari tiap-tiap alir:

1. Start
Adalah keadaan dimana program sudah dimulai.
2. Inisialisasi Variabel
Tahap pertama dalam membuat kode program ini biar bisa berjalan dengan baik maka dilakukanlah inisialisasi variabel serial agar dapat menghubungkan antara *master write* dan *slave receiver* dari perangkat keras yang telah dirancang dan inisialisasi variabel yang diperlukan dalam program.
3. Load dan Konfigurasi DMP
Setelah melakukan inisialisasi komunikasi serial maka kode program perlu load data dari DMP dan menkonfigurasi DMP agar bisa menghasilkan data yang diinginkan.
4. DMP ready
Disini adalah kondisi dimana jika DMP sudah *ready* maka akan menuju ke langkah selanjutnya, dan jika tidak maka program akan *looping* kembali ke load dan konfigurasi DMP.
5. Output *yaw*, *pitch* dan *roll*.
Menampilkan data nilai *pitch*, *roll*, *yaw*.
6. Konversi Nilai *Pitch*, *Roll*, *Yaw*
Setelah menampilkan data output *pitch*, *roll*, *yaw*. Maka nilai tersebut perlu di konversi agar bisa diimplementasikan pada *library mouse*.
7. Output *yaw* stabil?
Disini adalah kondisi dimana jika output nilai *yaw* stabil maka dapat dilakukan ke proses selanjutnya.
8. *Library Mouse Enabled*
Setelah mendapatkan data *pitch*, *roll*, *yaw* dan berhasil dikonversi maka pada tahap ini adalah mengaktifkan *library mouse* untuk menghasikan output gerakan pada laptop sebagai *mouse* dalam menggerakkan game Bus Simulator atau bisa disebut dengan *view controller*.
9. *Fifocount* == 1024
Disini adalah keadaan jika data dari DMP melebihi dari *packetSize* dari DMP (default) atau dihitung dalam *fifocount* sama dengan 1024 maka akan melakukan tugas yaitu men-*reset* *fifocount* kemudian program berhenti.
10. End
Program selesai.

5.3.2 Perancangan kode program konversi *yaw*, *pitch* dan *roll*

Pada perancangan konversi nilai *pitch*, *roll* dan *yaw* juga menggunakan diagram alir untuk menjelaskannya. Tahapan alurnya dapat dijelaskan seperti pada diagram alir dibawah ini.



Gambar 5.6 Diagram alir kode program konversi *yaw*, *pitch* dan *roll*

Diagram alir pada Gambar 5.6, dijelaskan tahapan alur proses dalam mengkonversi data dari *digital motion processing* (DMP) yang berupa *pitch*, *roll*, *yaw*. Proses konversi diawali dengan inisialisasi variable untuk menyimpan data pembacaan sensor. Dalam pengambilan data dilakukan pembacaan sensor setiap gerakan menoleh ke kanan dan ke kiri mendapatkan data nilai *pitch*, *roll*, *yaw*. Dari hasil pembacaan sensor yang berupa data bertipe *float* akan diproses untuk mendapatkan nilai minimum dari *pitch*, dan *yaw*. Berikut proses yang dijalankan:

```

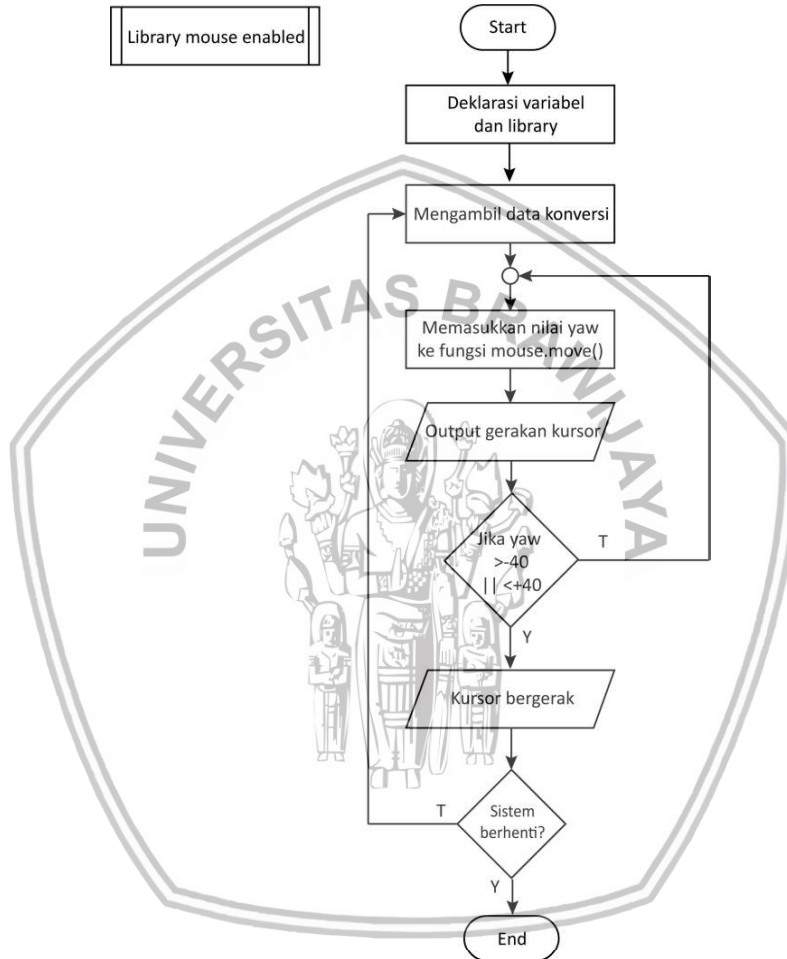
vertValue = pitch - vertZero
horzValue = yaw - horzZero
vertZero = pitch
horzZero = yaw
    
```

Didalam proses tersebut nilai *vertZero* dan *HorzZero* adalah hasil dari pemanggilan data dari *pitch* dan *yaw* yang mana nilai tersebut akan mengurangi nilai asli dari *pitch* dan *yaw* dengan inisialisasi dari keduanya adalah dengan nilai 0.0. kemudian didapatkan nilai minimum mendekati 0.00 yang kemudian dikalikan dengan sensitifitas yang telah ditentukan dari hasil *trial* dan *error*. Setelah diketahui hasilnya maka data hasil konversi dapat diimplementasikan pada *library mouse*.

Dengan adanya konversi agar data nilai *pitch* dan *yaw* yang diimplementasikan terhadap *library mouse* tidak terlalu besar dan menjadikan gerakan *mouse* akan tidak stabil. Dengan adanya konversi membuat nilai *yaw* dapat diimplementasikan sesuai dengan gerakan menoleh ke kanan dan ke kiri secara stabil dan nilai *pitch* yang digunakan juga stabil.

5.3.3 Perancangan kendali kursor

Pada perancangan kendali kursor dengan *library mouse* Arduino ini dijelaskan dengan menggunakan diagram alir seperti dibawah ini.



Gambar 5.7 Diagram alir kode program *library mouse*

Pada Gambar 5.7, dijelaskan *library mouse* memungkinkan mikrokontroler berbasis 32u4 untuk mengontrol gerakan kursor pada komputer yang terhubung melalui port USB dari laptop dengan menggunakan fungsi-fungsi yang ada pada *library* tersebut. Pada *library mouse* fungsi yang digunakan pada kode program utama adalah "*Mouse.move()*".

Pada fungsi "*Mouse.move()*" akan dapat menggerakkan kursor pada laptop melalui perintah yang dilakukan oleh Arduino dari hasil output sensor MPU9250. Sebelum menggunakan fungsi ini diharuskan menguji output dari sensor

MPU9250 terlebih dahulu dengan cara menggunakan perintah “Serial.print()”. Dengan cara ini adalah agar dapat meyakinkan terhadap nilai yang dihasilkan dari sensor MPU9250.

Parameter dari fungsi “*Mouse.move()*” yang diperlukan pada kode program utama adalah sebagai berikut:

1. xVal : adalah nilai untuk bergerak sepanjang sumbu X.
2. yVal : adalah nilai untuk bergerak sepanjang sumbu Y.

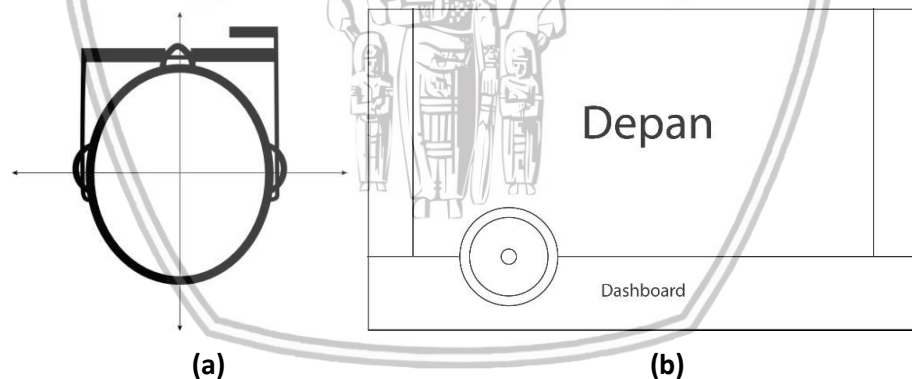
Sebelum menggunakan fungsi “*Mouse.move()*” adalah dengan menginisialisasi *library mouse* dengan syntax “*#include <Mouse.h>*” kemudian memanggil fungsi “*Mouse.begin()*” untuk memulai kontrol dari kode program utama.

5.4 Perancangan Prosedur Penggunaan Aplikasi pada Sistem

Pada perancangan ini peneliti menjelaskan bagaimana prosedur penggunaan aplikasi pada sistem yang akan digunakan. Prosedur yang dijelaskan adalah perancangan prosedur penggunaan pada game bus simulator dan prosedur penggunaan aplikasi *stream* dari laptop ke *smartphone*.

5.4.1 Perancangan prosedur penggunaan game bus simulator untuk user

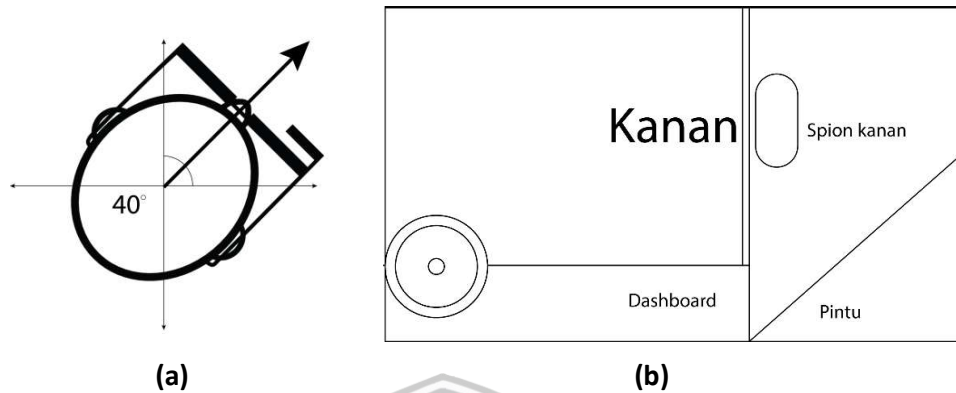
Pada perencaan ini peneliti menilustrasikan bagaimana penggunaan *user* terhadap game bus simulator dengan beberapa kondisi ketika menghadap ke depan, menoleh ke kanan, dan ke kiri pada saat memainkan game bus simulator.



Gambar 5.8 (a) penggunaan sistem pada saat menghadap ke depan (b) ilustrasi penggunaan game bus simulator ketika menghadap ke depan

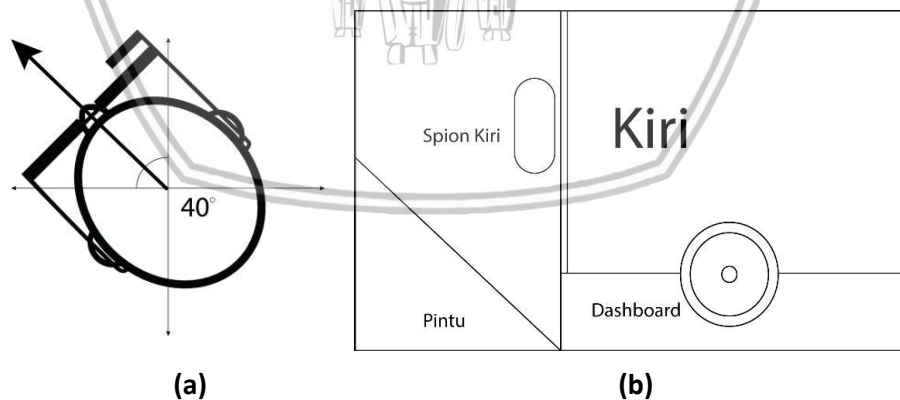
Pada Gambar 5.8, dijelaskan prosedur penggunaan game bus simulator dengan posisi kepala pengguna pada saat keadaan diam atau menghadap ke depan. Dengan posisi kepala pengguna yang menghadap ke depan maka di dalam game bus simulator juga akan memposisikan posisinya dengan menghadap ke depan atau bisa dikatakan menghadap kaca bagian depan *dashboard* pada game bus simulator. Pada posisi ini pengguna memposisikan keadaan kode program yang dijelaskan dimana pada saat kondisi *HorzValue* tidak sama dengan 0 maka

kursor pada *mouse* bergerak jika tidak maka kursor berhenti. Dengan kondisi diam menghadap ke depan maka kode program membaca kondisi HorzValue adalah sama dengan 0.



Gambar 5.9 (a) penggunaan sistem pada saat menghadap ke kanan (b) ilustrasi penggunaan game bus simulator ketika menghadap ke kanan

Pada Gambar 5.9, dijelaskan prosedur penggunaan game bus simulator dengan posisi kepala pengguna pada saat keadaan menoleh ke kanan dengan perkiraan 40 derajat. Dengan posisi kepala pengguna yang menoleh ke kanan maka di dalam game bus simulator juga akan memposisikan posisinya menoleh ke kanan atau bisa dikatakan dengan menghadap spion bagian kanan pada game bus simulator. Pada posisi ini pengguna memposisikan keadaan kode program yang dijelaskan dimana pada saat kondisi HorzValue tidak sama dengan 0 maka kursor pada *mouse* bergerak dan jika tidak maka kursor berhenti. Dengan kondisi bergerak menoleh ke kanan maka kode program membaca kondisi HorzValue adalah tidak sama dengan 0. Sampai pengguna berhenti untuk menoleh ke kanan dengan posisi perkiraan 40 derajat maka kode program membaca kondisi HorzValue adalah sama dengan 0 dan berhenti pada posisi tersebut.



Gambar 5.10 (a) penggunaan sistem pada saat menghadap ke kiri (b) ilustrasi penggunaan game bus simulator ketika menghadap ke kiri

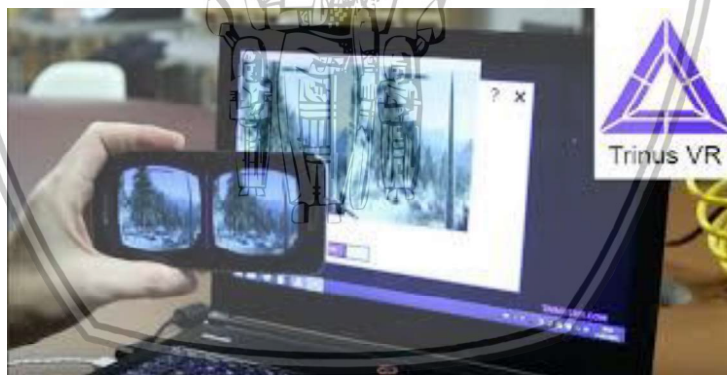
Pada Gambar 5.10, dijelaskan prosedur penggunaan game bus simulator dengan posisi kepala pengguna pada saat keadaan menoleh ke kiri dengan perkiraan 40 derajat. Dengan posisi kepala pengguna yang menoleh ke kiri maka

di dalam game bus simulator juga akan memposisikan posisinya menoleh ke kiri atau bisa dikatakan dengan menghadap spion bagian kiri pada game bus simulator. Pada posisi ini pengguna memposisikan keadaan kode program yang dijelaskan dimana pada saat kondisi HorzValue tidak sama dengan 0 maka kursor pada *mouse* bergerak dan jika tidak maka kursor berhenti. Dengan kondisi bergerak menoleh ke kiri maka kode program membaca kondisi HorzValue adalah tidak sama dengan 0. Sampai pengguna berhenti untuk menoleh ke kiri dengan posisi perkiraan 40 derajat maka kode program membaca kondisi HorzValue adalah sama dengan 0 dan berhenti pada posisi tersebut.

Hubungan dari beberapa prosedur penggunaan tersebut diatas adalah bahwa kode program yang dirancang dapat membaca nilai dari pergerakan pengguna yang telah ditentukan dan mengimplementasikan sesuai dengan posisi diam menghadap ke depan, menoleh ke kiri, dan menoleh ke kanan. Pada posisi menoleh ke kanan dan ke kiri pengguna dibatasi dengan menoleh sebesar perkiraan 40 derajat yang mana dalam posisi ini adalah batas dari game bus simulator pada saat menoleh ke kanan maupun ke kiri dengan sensitifitas dari game yaitu 1x.

5.4.2 Perancangan prosedur penggunaan aplikasi stream dari laptop ke *smartphone*

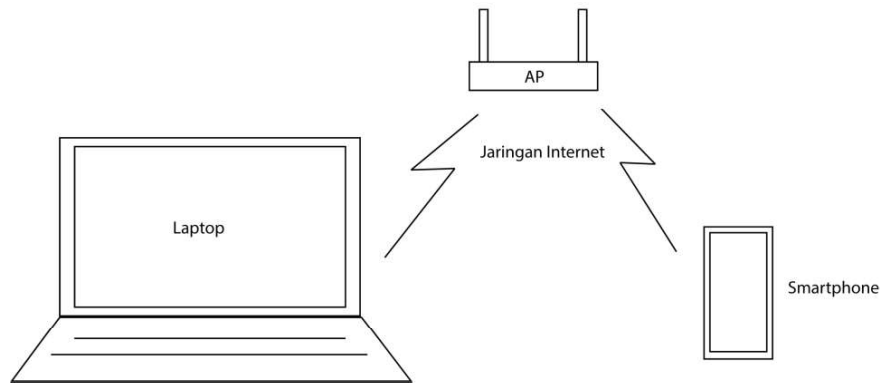
Pada perencaan ini peneliti menggunakan aplikasi *streaming* yang bernama Trinus VR. Aplikasi ini membantu peneliti untuk *streaming* game Bus Simulator dari laptop kepada *smartphone* dengan memanfaatkan koneksi internet.



Gambar 5.11 Aplikasi Trinus VR

Prosedur penggunaan dari aplikasi ini membutuhkan beberapa kebutuhan diantaranya adalah sebagai berikut :

1. *Smartphone* dengan CPU setidaknya Snapdragon 800.
2. PC atau laptop dengan minimum spesifikasi *graphics processing unit* (GPU) gaming.
3. Router atau USB *Tethering*.



Gambar 5.12 Penggunaan aplikasi Trinus VR

Pada Gambar 5.12, dijelaskan bahwa aplikasi yang sudah terinstall pada laptop maupun *smartphone* akan terhubung jika keduanya memiliki koneksi internet pada akses poin yang sama. Maka pada prosedur penggunaan aplikasi Trinus VR ini peneliti menggunakan *tethering* internet dari *smartphone* yang telah disediakan untuk kebutuhan penggunaan perangkat lunak aplikasi Trinus VR.

5.5 Implementasi Sistem

Pada bagian ini dijelaskan proses implementasi sistem dari implementasi perangkat keras sensor MPU9250, mikrokontroler dan implementasi perangkat lunak yang digunakan pada sistem.

5.5.1 Implementasi alat dan peletakan *view controller*

Sesuai dengan perancangan perangkat keras *view controller*, implementasi *view controller* ini terdiri atas mikrokontroler berbasis ATmega32U4 untuk sistem ini menggunakan Arduino Pro Mikro sebagai perangkat pemroses data. Sensor MPU9250 sebagai perangkat yang dapat membaca nilai masukan gerakan yang dihasilkan oleh *user*, serta dilengkapi dengan media komunikasi menggunakan kabel serial.

Pada Gambar 5.13 merupakan implementasi alat *view controller* terhadap pengguna yang diletakkan pada kepala dengan tampak depan dan tampak samping. Dengan menggunakan headband karet yang nyaman dapat memudahkan pengguna untuk memakainya secara fleksibel sesuai dengan ukuran kepala pengguna yang akan diuji dalam penelitian ini.



(a)

(b)

Gambar 5.13 (a) Pemakaian perangkat *view controller* tampak depan (b) Pemakaian perangkat *view controller* tampak depan

Pada Gambar 5.13, dijelaskan bahwa perangkat *view controller* ini menggunakan bahan *virtual box* (VR) dari plastic yang ringan sehingga membuat pemakaian terhadap pengguna menjadi nyaman. Sedangkan untuk implementasi sensor MPU9250 dan mikrokontroler Arduino pro mikro adalah dengan meletakkan ke bagian samping dan depan dari *virtual box* (VR). Bagian samping untuk perangkat mikrokontroler Arduino pro mikro, dan bagian depan adalah untuk sensor MPU9250. Jarak antara Arduino pro mikro dengan sensor MPU9250 adalah dengan menyesuaikan panjang kabel *jumper*. Peletakan sensor MPU9250 pada posisi tengah adalah bertujuan untuk memposisikan pengambilan nilai gerakan dari sensor MPU9250, dengan posisi yang ada di bagian tengah *virtual box* (VR) akan memudahkan pemacaan sensor pada saat bermain game bus simulator.

5.5.2 Implementasi perangkat keras

Dalam melakukan tahap implementasi perangkat keras harus disesuaikan pada perancangan perangkat keras yang telah di buat pada sub bab 5.1.3. Sistem menggunakan sensor akselerometer MPU9250 dan Mikrokontroler Arduino Pro Mikro. Sensor Akselerometer MPU9250 digunakan untuk mendapatkan input berupa gerakan dan kemudian inputan tersebut akan dikirim ke Mikrokontroler Arduino Pro Mikro untuk diproses menjadi output berupa gerakan *mouse* pada laptop yang akan digunakan untuk menggerakkan game bus simulator sebagai *view controller*. Berikut adalah hasil dari implementasi perangkat keras pada Gambar 5.14.



Gambar 5.14 Perangkat *view controller* tampak depan, samping, dan atas

Konfigurasi pin pada seluruh perangkat sesuai dengan perancangan pada Gambar 5.2 dan masing-masing komponen terhubung dengan kabel *jumper*. Pada Gambar 5.14 menunjukkan hasil implementasi dari perancangan *view controller*.

5.5.3 Implementasi perangkat lunak

Pada sub-bab implementasi perangkat lunak terdapat beberapa bagian sub-bab program yaitu program pembacaan data sensor akselerometer MPU9250 dan program penggunaan *library mouse* dengan data sensor yang dihasilkan oleh MPU9250. Program menggunakan Bahasa C pada Arduino IDE.

5.5.3.1 Implementasi pembacaan data sensor MPU9250

Implementasi kode program pada sistem *view controller* ini memiliki 2 fungsi utama. Yang pertama adalah kode program yang digunakan untuk memperoleh data pengukuran gerakan dari sensor MPU9250. fungsi yang kedua yaitu menjadikan hasil dari pembacaan *digital motion processor* (DMP) perangkat *mouse* sebagai sistem untuk *view controller* pada game bus simulator dengan menggunakan *library mouse* "*mouse.h*". pada sub bab ini peneliti menjelaskan fungsi yang pertama.

Dalam penelitian ini informasi *output* data *digital motion processor* (DMP) dari pengukuran gerakan sensor MPU9250 cukup kuat untuk pengaplikasian sebagai *mouse* pada laptop. Untuk mendapatkan informasi data *digital motion processor* (DMP) dibutuhkan file pustaka yang mendefinisikan *library* dari sensor MPU9250 atau sejenisnya yang mendukung untuk pembacaan *digital motion processor* (DMP). Pada penelitian ini menggunakan file *library* dari MPU6050.

Tabel 5.2 Mendefinisikan *library*

No	Kode Program
1	#include "I2Cdev.h"
2	#include <Mouse.h>
3	#include "MPU6050_6Axis_MotionApps20.h"
4	#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
5	#include "Wire.h"

Setelah mendefinisikan *library* yang dibutuhkan, kemudian mendeklarasikan objek, menginisialisasi, dan fungsi dari *digital motion processor* (DMP).

Tabel 5.3 Mendeklarasikan objek, menginialisasi fungsi DMP

No	Kode Program
1	MPU6050 mpu;
2	Quaternion q; // [w, x, y, z] quaternion
3	container

Tabel 5.3 Mendeklarasikan objek, menginialisasi fungsi DMP (Lanjutan)

No	Kode Program
4	VectorFloat gravity; // [x, y, z] gravity vector
5	float euler[3]; // Euler angle container
6	float ypr[3]; // [yaw, pitch, roll]
7	float a, b, c; //pitch roll yaw
8	float vertZero, horzZero;
9	float vertValue, horzValue;
10	
11	const int sensitivity = 10;
12	
13	bool isStabil = false;
14	float initC, prevC = 0;
15	
16	Mpu.initialize();
17	devStatus = mpu.dmpInitialize();

Pada Tabel 5.3, dijelaskan mendekarasikan variabel yang akan digunakan, Jika sensor MPU9250 berfungsi dengan baik, informasi orientasi dari *digital motion processor* (DMP) dapat dibaca dan dihitung untuk menghasilkan nilai yang sesuai untuk kebutuhan penggunaan *library mouse* sebagai *cursor* pada laptop. Pada baris sensitifitas digunakan untuk memberikan pengaruh terhadap keluaran dari konversi yang akan dilakukan untuk pergerakan *cursor mouse* stabil.

Tabel 5.4 Kode program untuk mendapatkan nilai dari sensor

No	Kode Program
1	mpu.dmpGetQuaternion(&q, fifoBuffer);
2	mpu.dmpGetGravity(&gravity, &q);
3	mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
4	a = ypr[1] /PI * 180; //pitch
5	b = ypr[2] /PI * 180; //roll
6	c = ypr[0] /PI * 180; //yaw
7	Serial.print("pry\t");
8	Serial.print(a);
9	Serial.print("\t");
10	Serial.print(b);
11	Serial.print("\t");
12	Serial.println(c);

Pada Tabel 5.4, dijelaskan yaitu pada baris pertama kode program adalah mendapatkan nilai *fifoBuffer* yang didapatkan oleh *digital motion processing* (DMP). Kemudian pada baris kedua adalah kode program mendapatkan nilai *gravity* yaitu nilai vektor gravitasi yang dihasilkan oleh *digital motion processing* (DMP). Kemudian pada baris ketiga kode program akan mendapatkan nilai *yaw*, *pitch*, dan *roll* yang akan ditampilkan ke dalam array dari parameter “ypr”, lalu pada baris 4 sampai 6 adalah menampilkan nilai sudut dalam derajat dengan rumus hasil dari nilai yang didapatkan oleh *digital motion processing* (DMP) yaitu *yaw*, *pitch*, dan *roll* akan dibagi dengan hasil perkalian antara PI dengan 180. Pada baris 7 sampai dengan baris 12 menjelaskan output dari nilai *yaw*, *pitch*, dan *roll* yang akan ditampilkan pada serial monitor.

5.5.3.2 Implementasi mekanisme penggunaan *library mouse.h* dan konversi nilai dari sensor MPU9250

Implementasi kode program dari sistem *view controller* ini memiliki fungsi yang kedua yaitu menjadikan hasil dari pembacaan *digital motion processor* (DMP) perangkat *mouse* sebagai sistem untuk *view controller* pada game bus simulator dengan menggunakan *library mouse* “*mouse.h*”.

Tabel 5.5 Kode program inisialisasi *library mouse*

No	Kode Program
1	#include “mouse.h”

Pada Tabel 5.5, menjelaskan inisialisasi *library mouse.h*. Setelah mendefinisikan *library mouse*, data dari *digital motion processor* (DMP) perlu di konversikan kedalam gerakan *mouse* untuk menghasilkan sistem *view controller*.

Tabel 5.6 Kode program untuk mendapatkan konversi data dan menggerakkan *mouse*

No	Kode Program
1	blinkState = !blinkState;
2	digitalWrite(LED_PIN, blinkState);
3	vertValue = a - vertZero; //a : pitch
4	horzValue = c - horzZero; //c : yaw
5	vertZero = a;
6	horzZero = c;
7	if (((vertValue != 0) (horzValue != 0)) && isStabil &&
8	(c < initC+40 && c > initC-40)){
9	Mouse.move(0, -vertValue * sensitivity, 0);
10	Mouse.move(horzValue * sensitivity, 0, 0);
11	}

Pada Tabel 5.6, dijelaskan bahwa hasil pembacaan sensor MPU9250 yaitu nilai *pitch* dan *yaw* akan dikonversikan dengan mengurangkan dengan nilai *yaw* dan *pitch* sendiri dan jika nilai dari “horzValue” tidak sama dengan 0, maka akan memanggil perintah “*Mouse.move()*” untuk menggerakkan kursor pada laptop dengan nilai dari pembacaan sensor MPU9250. Pada kode program baris 8 dan 9 ada perintah “*sensitivity*” yaitu berfungsi untuk mengatur kecepatan dari implementasi kursor pada laptop biar pergerakan pengguna dengan kursor seimbang.

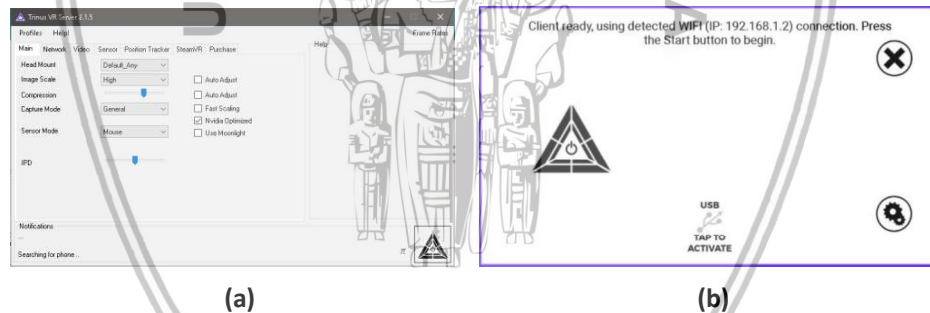
Setelah semua kode program yang dibutuhkan sudah terpenuhi maka implementasi kode program sudah selesai serta sistem *view controller* dapat diimplementasikan.

5.5.4 Implementasi prosedur penggunaan aplikasi

Pada sub-bab implementasi prosedur penggunaan aplikasi ini akan menjelaskan implementasi terhadap aplikasi yang digunakan pada sistem *view controller*.

5.5.4.1 Implementasi prosedur penggunaan aplikasi trinus VR sebagai *streaming*

Implementasi aplikasi Trinus VR sebagai *streaming* video adalah dengan menginstal aplikasinya yang ada pada laptop dan *smartphone*. Masing-masing di aplikasi dapat didapatkan pada situs resmi *website* dari Trinus VR untuk penggunaan pada laptop dan pada *google play store* untuk penggunaan pada *smartphone*.



Gambar 5.15 (a) Aplikasi trinus VR pada laptop (b) Aplikasi trinus VR pada *smartphone*

Setelah semua aplikasi Trinus VR sudah terinstal pada perangkat laptop dan *smartphone* maka untuk menghubungkannya adalah dengan cara menggunakan koneksi internet pada jaringan yang sama dengan mengaktifkan tombol bergambar segitiga pada aplikasi Trinus VR yang ada di laptop maupun di *smartphone*. Setelah keduanya mengkonfigurasi koneksi, maka akan mendapatkan respon dari *smartphone* kalau aplikasi sudah terhubung.

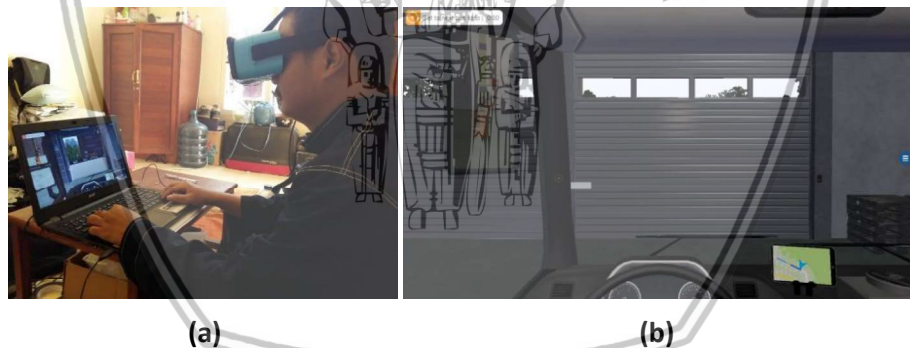


Gambar 5.16 Aplikasi trinus VR pada *smartphone* yang sudah terkoneksi

Pada Gambar 5.16, dijelaskan bahwa hasil *streaming* dari laptop ke *smartphone* dengan menggunakan aplikasi Trinus VR adalah dengan langsung menampilkan hasil *streaming* berupa pembagian gambar menjadi dua. Fungsi dari pembagian tersebut adalah sebagai implementasi dari perangkat keras *View Controller* yang menggunakan *Virtual Box* (VR) dengan lensa cembung yang sudah terpasang pada *Virtual Box* (VR).

5.5.4.2 Implementasi prosedur penggunaan game bus simulator

Implementasi penggunaan game bus simulator dijelaskan ketika kondisi dari *user* saat memainkan game bus simulator pada saat diam atau melihat kedepan, menoleh ke kiri, dan menoleh ke kanan. Adapun implementasinya seperti pada gambar dibawah ini:



Gambar 5.17 (a) *user* menggunakan *view controller* dalam posisi melihat ke depan (b) posisi didalam game bus simulator

Pada Gambar 5.17(a), dijelaskan pada saat *user* memainkan game bus simulator dengan keadaan posisi diam atau melihat ke depan maka didalam game bus simulator juga memposisikan keadaan yang sama yaitu dengan melihat ke depan. Pergerakan *user* terhadap game bus simulator dijalankan berdasarkan input sensor MPU9250 dan diproses kemudian diimplementasikan menjadi gerakan kursor. Pada Gambar 5.17(b), pengguna memposisikan keadaan kode program yang dijelaskan dimana pada saat kondisi *HorzValue* tidak sama dengan 0 maka kursor pada *mouse* bergerak dan jika tidak maka kursor berhenti.

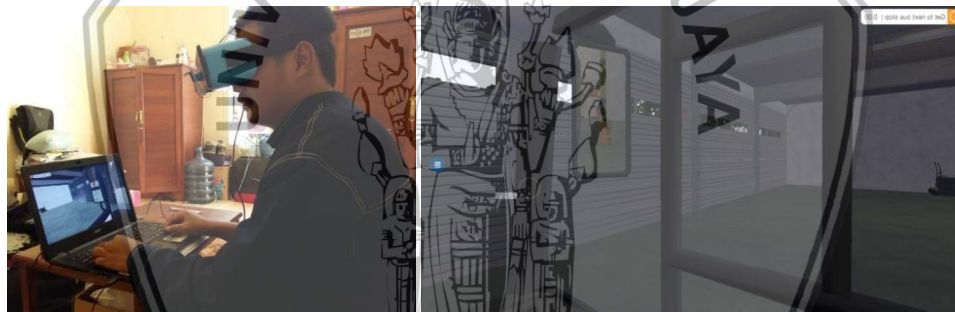


(a)

(b)

Gambar 5.18 (a) user menggunakan *view controller* dalam posisi melihat ke kanan (b) posisi didalam game bus simulator

Pada Gambar 5.15(a), dijelaskan pada saat *user* memainkan game bus simulator dengan keadaan posisi menoleh ke kanan maka didalam game bus simulator juga memposisikan keadaan yang sama yaitu dengan menoleh ke kanan. Pergerakan *user* terhadap game bus simulator dijalankan berdasarkan input sensor MPU9250 dan diproses kemudian diimplementasikan menjadi gerakan kursor. Pada Gambar 5.15(b), pengguna memposisikan keadaan kode program yang dijelaskan dimana pada saat kondisi *HorzValue* tidak sama dengan 0 maka kursor pada *mouse* bergerak.



(a)

(b)

Gambar 5.19 (a) user menggunakan *view controller* dalam posisi melihat ke kiri (b) posisi didalam game bus simulator

Pada Gambar 5.19(a), dijelaskan pada saat *user* memainkan game bus simulator dengan keadaan posisi menoleh ke kiri maka didalam game bus simulator juga memposisikan keadaan yang sama yaitu dengan menoleh ke kiri. Pergerakan *user* terhadap game bus simulator dijalankan berdasarkan input sensor MPU9250 dan diproses kemudian diimplementasikan menjadi gerakan kursor. Pada Gambar 5.16(b), pengguna memposisikan keadaan kode program yang dijelaskan dimana pada saat kondisi *HorzValue* tidak sama dengan 0 maka kursor pada *mouse* bergerak.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini membahas tentang tahap pengujian dan analisis setelah dilakukannya implementasi pada sistem. Pengujian pada sistem meliputi pengujian terhadap sensor MPU9250, pengujian fungsional sistem dan pengujian korespondensi.

6.1 Pengujian Sensor MPU9250

Pada sub bab ini akan dilakukan pengujian terhadap sensor MPU9250

6.1.1 Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui apakah sensor MPU9250 mampu membaca perubahan sudut pada setiap sumbu x, y dan z.

6.1.2 Prosedur Pengujian

Agar pengujian ini berjalan dengan lancar dalam pelaksanaannya maka pengujian ini dilakukan sesuai dengan prosedur sebagai berikut:

1. Siapkan sensor yang akan digunakan yaitu sensor MPU9250 yang telah dihubungkan pada Arduino pro mikro, kemudian hubungkan Arduino pro mikro dengan laptop menggunakan kabel serial USB.
2. Jalankan Serial Monitor pada program Arduino IDE.
3. Tempatkan sensor pada posisi datar dan secara bertahap ubah posisi sensor ke posisi yang ditentukan.

6.1.3 Pelaksanaan pengujian

Pengujian ini dilakukan berdasarkan prosedur yang telah dijelaskan sebelumnya. Pengujian ini akan dilakukan pada sudut Y dan X.

6.1.4 Hasil Pengujian

Setelah dilakukan pengujian terhadap sensor MPU9250 maka akan didapatkan hasil sebagai berikut:

1. Pengujian sumbu X .

Tabel 6.1 Hasil pengujian pada sumbu X

No	Sudut (Derajat)	Nilai keluaran sensor (roll)	Deviasi	Sudut (Derajat)	Nilai keluaran sensor (roll)	Deviasi
1	45	45.16	-0.16	-45	-45.10	0.10
2	45	45.20	-0.2	-45	-45.17	0.17
3	45	45.36	-0.36	-45	-45.21	0.21
4	45	45.10	-0.1	-45	-45.11	0.11
5	45	45.18	-0.18	-45	-45.19	0.19
6	45	45.09	-0.09	-45	-45.23	0.23
7	45	45.21	-0.21	-45	-45.29	0.29

Tabel 6.1 Hasil pengujian pada sumbu X (lanjutan)

No	Sudut (derajat)	Nilai keluaran sensor (roll)	Deviasi	Sudut (derajat)	Nilai keluaran sensor (roll)	Deviasi
8	45	45.30	-0.3	-45	-45.09	0.09
9	45	45.17	-0.17	-45	-45.18	0.18
10	45	45.07	-0.07	-45	-45.08	0.08
	RMSE		0.2033	RMSE		0.1775

Berdasarkan Tabel 6.1, didapatkan nilai RMSE dari hasil pengujian sensor pada sumbu X dengan sudut 45 derajat sebesar 0.2033 derajat dan dari sudut -45 derajat mendapatkan RMSE sebesar 0.1775 derajat.

2. Pengujian sumbu Y.

Tabel 6.2 Hasil pengujian pada sumbu Y

No	Sudut (derajat)	Nilai keluaran sensor (pitch)	Deviasi	Sudut (derajat)	Nilai keluaran sensor (pitch)	Deviasi
1	45	45.30	-0.3	-45	-45.16	0.16
2	45	45.20	-0.2	-45	-45.25	0.25
3	45	45.11	-0.11	-45	-45.16	0.16
4	45	45.28	-0.28	-45	-45.19	0.19
5	45	45.31	-0.31	-45	-45.20	0.2
6	45	45.01	-0.01	-45	-45.29	0.29
7	45	45.25	-0.25	-45	-45.30	0.3
8	45	45.38	-0.38	-45	-45.08	0.08
9	45	45.15	-0.15	-45	-45.28	0.28
10	45	45.27	-0.27	-45	-45.05	0.05
	RMSE		0.2487	RMSE		0.2124

Berdasarkan Tabel 6.2, didapatkan nilai RMSE dari hasil pengujian sensor pada sumbu Y dengan sudut 45 derajat sebesar 0.2487 derajat dan dari sudut -45 derajat mendapatkan RMSE sebesar 0.2124 derajat.

3. Pengujian Sumbu Z.

Tabel 6.3 Hasil pengujian pada sumbu Y

No	Sudut (derajat)	Nilai keluaran sensor (yaw)	Deviasi	Sudut (derajat)	Nilai keluaran sensor (yaw)	Deviasi
1	45	45.10	-0.1	-45	-45.11	0.11
2	45	45.30	-0.3	-45	-45.05	0.05
3	45	45.01	-0.01	-45	-45.18	0.18
4	45	45.21	-0.21	-45	-45.26	0.26
5	45	45.11	-0.11	-45	-45.02	0.02

Tabel 6.3 Hasil pengujian pada sumbu Y (lanjutan)

No	Sudut (derajat)	Nilai keluaran sensor (yaw)	Deviasi	Sudut (derajat)	Nilai keluaran sensor (yaw)	Deviasi
6	45	45.07	-0.07	-45	-45.15	0.15
7	45	45.28	-0.28	-45	-45.20	0.2
8	45	45.22	-0.22	-45	-45.12	0.12
9	45	45.09	-0.09	-45	-45.29	0.29
10	45	45.20	-0.2	-45	-45.09	0.09
	RMSE		0.1833	RMSE		0.1685

Berdasarkan Tabel 6.3 didapatkan nilai RMSE dari hasil pengujian sensor pada sumbu Y dengan sudut 45 derajat sebesar 0.1833 derajat dan dari sudut -45 derajat mendapatkan RMSE sebesar 0.1685 derajat.

6.1.5 Analisis Hasil Pengujian

Berdasarkan pengujian pada tabel 6.1, 6.2 dan table 6.3, Sensor MPU9250 mampu membaca kondisi sudut sebenarnya dengan selisih dengan sudut sebenarnya dengan hasil pembacaan antara $\pm 0.2^\circ$. Hasil rekapitulasi RMSE dapat dilihat pada tabel dibawah ini:

Tabel 6.4 Hasil rekapitulasi RMSE

Sumbu	Sudut (derajat)	RMSE	Sudut (derajat)	RMSE
X	45	0.2033	-45	0.1775
Y	45	0.2487	-45	0.2124
Z	45	0.1833	-45	0.1685
	Rata-Rata	0.211767	Rata-rata	0.186133

Dengan selisih hasil nilai dari pembacaan sensor tersebut maka sensor dapat digunakan untuk *view controller*.

6.2 Pengujian Fungsional Sistem

Pengujian ini untuk mengetahui fungsional sistem mampu berjalan dengan baik sesuai kasus yang diberikan oleh peneliti.

6.2.1 Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui apakah sistem stabil jika perubahan sudut pada nilai *yaw* bertambah sebanyak lebih dari 40 dan lebih dari -40 (tengok kanan dan kiri).

6.2.2 Prosedur Pengujian

Agar pengujian ini berjalan dengan lancar dalam pelaksanaannya maka pengujian ini dilakukan sesuai dengan prosedur berikut:

1. Siapkan sensor MPU9250 yang telah dihubungkan pada Arduino, kemudian hubungkan Arduino dengan laptop menggunakan kabel usb.

2. Jalankan serial monitor pada program Arduino IDE.
3. Tempatkan sensor pada posisi datar dan secara bertahap ubah ke posisi yang ditentukan (tengok kanan dan kiri).

6.2.3 Pelaksanaan pengujian

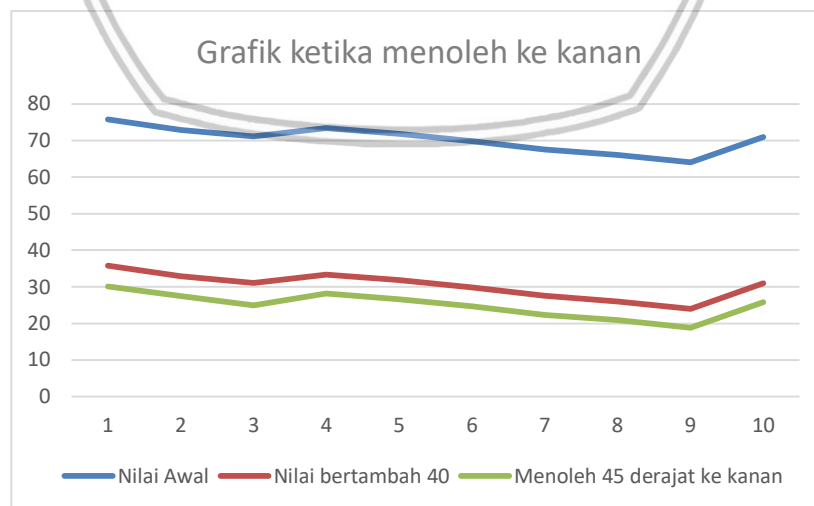
Pada pengujian fungsional sistem, proses pengambilan data adalah pada saat sistem berubah posisi dari titik awal menuju titik berhenti dan data yang diambil pada saat titik berhenti tersebut. Titik berhenti yang dimaksud adalah pada saat *user* tengok kanan dan kiri sesuai dengan keadaan pada game bus simulator.

6.2.4 Hasil Pengujian

Tabel 6.5 Hasil pengujian ketika menoleh ke kanan

Nilai awal (derajat)	Nilai menoleh ke kanan 45 (derajat)	Nilai ketika kembali ke tengah (derajat)	Berhenti saat nilai yaw lebih dari 40 (derajat)
81.4	126.6	81	Ya
83.2	128.5	83.4	Ya
86.5	131.62	86.52	Ya
87.22	132.39	87.16	Ya
88.4	133.71	88.51	Ya
90	135.11	89.9	Ya
89.92	135.18	90.15	Ya
86.47	131.69	86.58	Ya
85	130.06	84.95	Ya
87.2	132.36	87.06	Ya
HASIL			SUKSES

Berdasarkan Tabel 6.5, didapatkan hasil pembacaan sensor ketika *user* menoleh ke kanan dengan sudut 45 derajat dengan 10 kali percobaan.



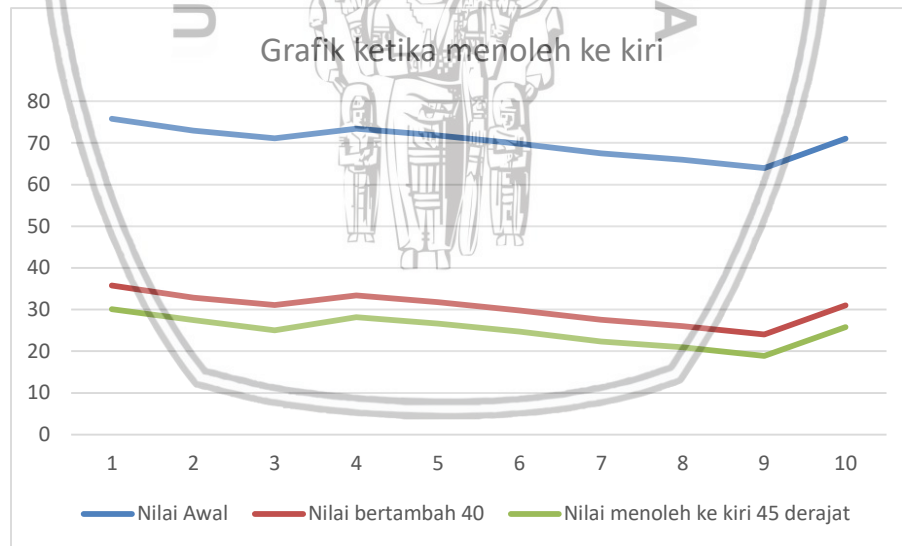
Gambar 6.1 Grafik hasil pengujian menoleh ke kanan pada sudut 45 derajat

Berdasarkan Gambar 6.1 terdapat tiga garis yaitu nilai awal, nilai bertambah 40 dan nilai menoleh 45 derajat ke kanan hasil pembacaan sensor ketika *user* menoleh ke kanan dengan sudut 45 derajat.

Tabel 6.6 Hasil pengujian ketika menoleh ke kiri

Nilai awal (derajat)	Nilai menoleh ke kanan 45 (derajat)	Nilai ketika kembali ke tengah (derajat)	Berhenti saat nilai yaw lebih dari 40 (derajat)
75.8	30.1	74	Ya
72.9	27.42	72.62	Ya
71.1	25	71.03	Ya
73.4	28.18	73.49	Ya
71.8	26.61	71.81	Ya
69.8	24.68	69.86	Ya
67.5	22.33	67.43	Ya
66	20.97	66.09	Ya
64	18.86	64.13	Ya
71	25.8	70.96	Ya
HASIL			SUKSES

Berdasarkan Tabel 6.6, didapatkan hasil pembacaan sensor ketika *user* menoleh ke kiri dengan sudut 45 derajat dengan 10 kali percobaan.



Gambar 6.2 Grafik hasil pengujian menoleh ke kiri pada sudut 45 derajat

Berdasarkan Gambar 6.2 terdapat tiga garis yaitu nilai awal, nilai bertambah 40 dan nilai menoleh 45 derajat ke kanan hasil pembacaan sensor ketika *user* menoleh ke kiri dengan sudut 45 derajat.

6.2.5 Analisis Hasil Pengujian

Berasarkan data hasil pengujian, sistem mampu menstabilkan sudut ketika nilai sistem lebih dari sudut yang ditentukan. Pada pegujian ini jika nilai sistem bertambah sebanyak lebih dari 40 dan - 40 maka kursor yang ada pada laptop akan berhenti pada posisi tersebut dan menghiraukan pergerakan jika nilai dari sistem bertambah lebih dari 40 dan -40.

6.3 Pengujian Korespondensi

Pengujian sistem korespondensi adalah pengujian terhadap sistem yang dilakukan oleh beberapa *user*.

6.3.1 Tujuan Pengujian

Pengujian sistem korespondensi dilakukan untuk mengetahui *feedback* dari *user* saat menggunakan sistem.

6.3.2 Prosedur Pengujian

Agar pengujian ini berjalan lancar dalam pelaksanaannya maka pengujian ini dilakukan sesuai dengan prosedur sebagai berikut:

1. Siapkan sistem *view controller*.
2. Pasangkan sistem *view controller* ke kepala *user*.
3. Hubungkan sistem *view controller* ke komputer.
4. *User* menggerakkan kepala untuk mengendalikan kursor pada game Bus Simulator.
5. *User* memberikan *feedback* dari kerja sistem *view controller*.

6.3.3 Pelaksanaan pengujian

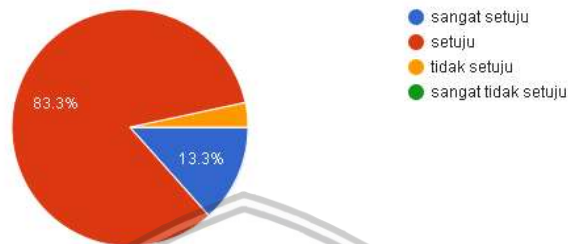
Pada pengujian sistem korespondensi, proses pengumpulan *feedback* dari sistem oleh *user* dilakukan oleh 30 *user* dengan ketentuan 15 *user* "*gamer*" dan 15 *user* "*non-gamer*". Proses pengumpulan *feedback* dari *user* dilakukan dengan cara memasang sistem *view controller* ke kepala *user* dan *user* akan menggunakan sistem *view controller* dengan memainkan game bus simulator dengan menoleh ke kanan dan ke kiri. Hasil *feedback* terhadap pertanyaan yang diterima oleh *user* akan terlampir pada Lampiran.

6.3.4 Hasil Pengujian

1. Pengujian kesesuaian

Apakah penggunaan view controller dalam bermain game bus simulator sudah sesuai dengan pergerakan kepala ketika melihat depan, menoleh ke kanan, dan ke kiri?

30 responses

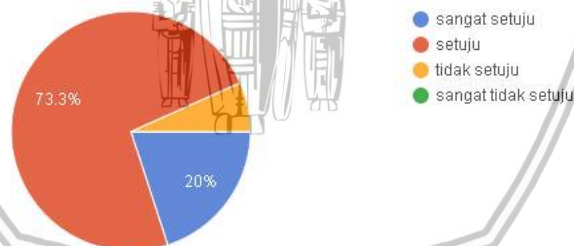


Gambar 6.3 diagram hasil umpan balik kesesuaian menggunakan *view controller*

Berdasarkan gambar 6.3 didapatkan persentasi *user* memilih setuju sebesar 83,3%, *user* memilih sangat setuju sebesar 13% dan *user* memilih tidak setuju 3,3%.

Apakah penggunaan view controller ini sudah bagus dibandingkan dengan menggunakan mouse?

30 responses

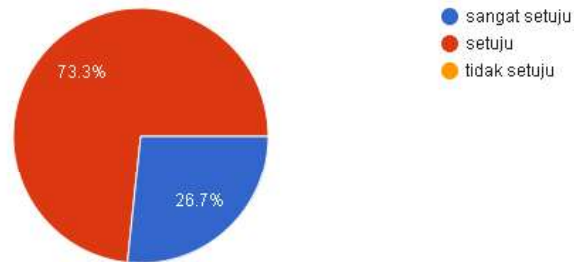


Gambar 6.4 diagram hasil umpan balik kesesuaian menggunakan *view controller*

Berdasarkan gambar 6.4 didapatkan persentasi *user* memilih setuju sebesar 73,3%, *user* memilih sangat setuju sebesar 20% dan *user* memilih tidak setuju 6,7%.

Apakah anda menyukai penggunaan view controller untuk bermain game Bus Simulator

80 responses



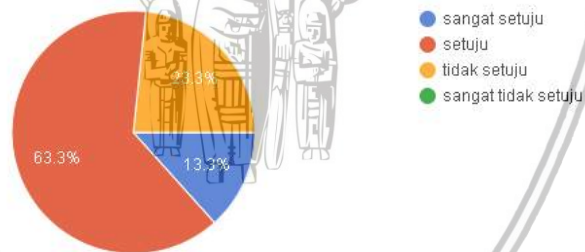
Gambar 6.5 diagram hasil umpan balik kesesuaian menggunakan *view controller*

Berdasarkan gambar 6.5 didapatkan persentasi *user* memilih setuju sebesar 73,3% dan *user* memilih sangat setuju sebesar 26,7%.

2. Pengujian Nilai

Pada saat penggunaan view controller dalam bermain Bus Simulator, apakah view controller sudah stabil?

80 responses

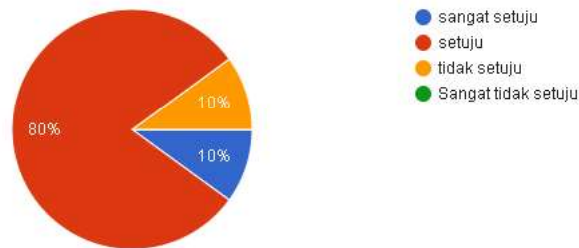


Gambar 6.6 diagram hasil umpan balik nilai menggunakan *view controller*

Berdasarkan gambar 6.6 didapatkan persentasi *user* memilih setuju sebesar 63,3%, *user* memilih sangat setuju sebesar 13,3% dan *user* memilih tidak setuju 23,3%.

Pada saat penggunaan view controller dalam bermain Bus Simulator, tingkat sensitifitas dari view controller tersebut sudah sesuai dengan batas sudut saat digunakan untuk menoleh ke kanan dan ke kiri

30 responses

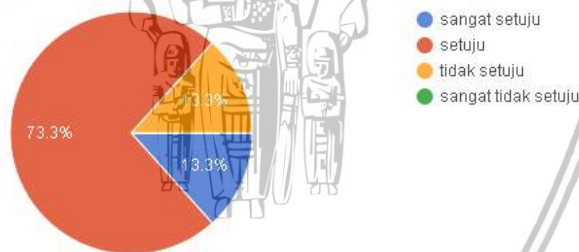


Gambar 6.7 diagram hasil umpan balik nilai menggunakan view controller

Berdasarkan gambar 6.7 didapatkan persentasi *user* memilih setuju sebesar 80%, *user* memilih sangat setuju sebesar 10% dan *user* memilih tidak setuju 10%.

Pada saat bermain game bus simulator, apakah view controller sudah akurat dalam penggunaannya?

30 responses



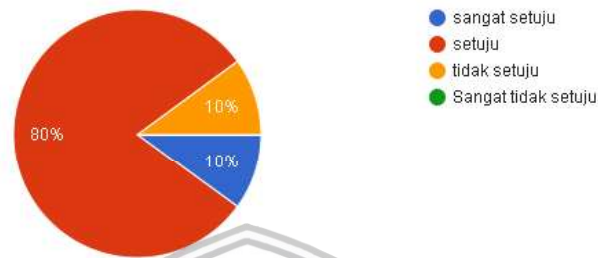
Gambar 6.8 diagram hasil umpan balik nilai menggunakan view controller

Berdasarkan gambar 6.8 didapatkan persentasi *user* memilih setuju sebesar 73,3%, *user* memilih sangat setuju sebesar 13,3% dan *user* memilih tidak setuju 13,3%.

3. Pengujian Kemudahan

Apakah dalam bermain game bus simulator dengan penggunaan View Controller ini mudah digunakan?

30 responses

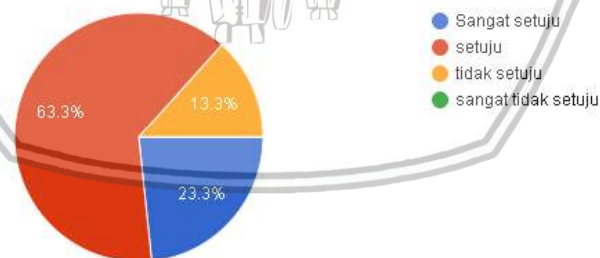


Gambar 6.9 diagram hasil umpan balik kemudahan menggunakan *view controller*

Berdasarkan gambar 6.9 didapatkan persentasi *user* memilih setuju sebesar 80%, *user* memilih sangat setuju sebesar 10% dan *user* memilih tidak setuju 10%.

Pada saat memainkan game bus simulator dengan menggunakan View Controller, apakah mudah saat mengendalikan pergerakan menoleh ke kiri dan ke kanan?

30 responses

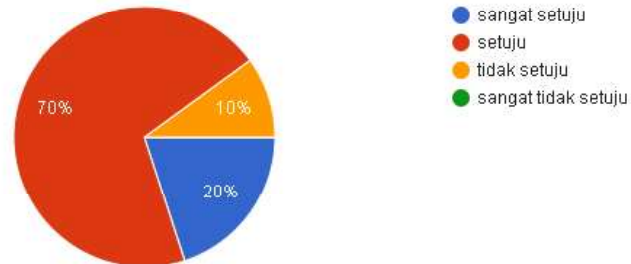


Gambar 6.10 diagram hasil umpan balik kemudahan menggunakan *view controller*

Berdasarkan gambar 6.10 didapatkan persentasi *user* memilih setuju sebesar 63,3%, *user* memilih sangat setuju sebesar 23,3% dan *user* memilih tidak setuju 13,3%.

apakah view controller sudah cocok untuk digunakan dalam bermain game bus simulator?

30 responses



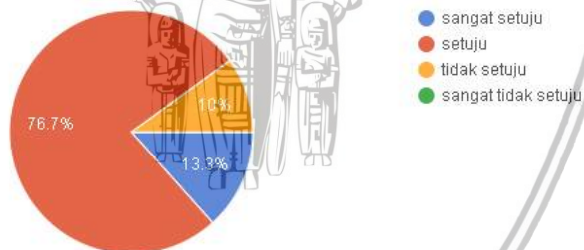
Gambar 6.11 diagram hasil umpan balik kemudahan menggunakan *view controller*

Berdasarkan gambar 6.11 didapatkan persentasi *user* memilih setuju sebesar 70%, *user* memilih sangat setuju sebesar 20% dan *user* memilih tidak setuju 10%.

4. Pengujian Perasaan

Apakah anda senang saat memainkan game bus simulator dengan menggunakan View Controller?

30 responses

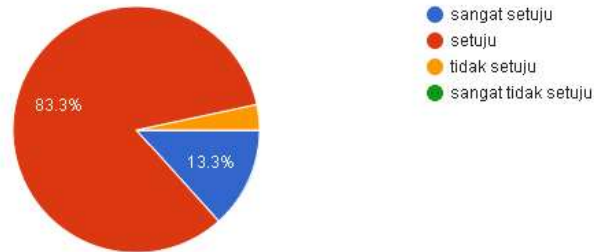


Gambar 6.12 diagram hasil umpan balik perasaan menggunakan *view controller*

Berdasarkan gambar 6.12 didapatkan persentasi *user* memilih setuju sebesar 76,7%, *user* memilih sangat setuju sebesar 13,3% dan *user* memilih tidak setuju 10%.

Apakah penggunaan view controller dalam bermain game bus simulator sudah baik saat digunakan?

30 responses

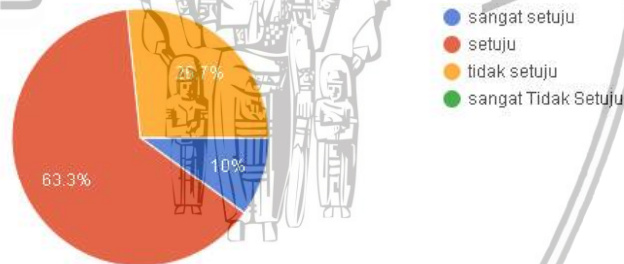


Gambar 6.13 diagram hasil umpan balik perasaan menggunakan *view controller*

Berdasarkan gambar 6.13 didapatkan persentasi *user* memilih setuju sebesar 83,3%, *user* memilih sangat setuju sebesar 13,3% dan *user* memilih tidak setuju 3,3%.

Apakah anda merasa nyaman saat menggunakan View Controller dalam bermain game bus simulator?

30 responses



Gambar 6.14 diagram hasil umpan balik perasaan menggunakan *view controller*

Berdasarkan gambar 6.14 didapatkan persentasi *user* memilih setuju sebesar 63,3%, *user* memilih sangat setuju sebesar 10% dan *user* memilih tidak setuju 26,7%.

6.3.5 Analisis Hasil Pengujian

Untuk menghitung data hasil pengujian korespondensi dengan menggunakan skala likert sebagai berikut.

$$\text{Rumus Indeks \%} = \frac{\text{Total Skor}}{Y} \times 100$$

Total Skor adalah nilai yang didapatkan dari jumlah data korensponden yang memilih antara sangat setuju, setuju, tidak setuju dan sangat tidak setuju.

Setiap pilihan akan memiliki nilai tersendiri mulai dari 1 sampai 4. Y adalah skor tertinggi pada likert dalam hal ini adalah 4.

Selain membutuhkan rumus indeks, perhitungan ini juga membutuhkan rumus interval untuk membagi kategori dari nilai yang didapat ke dalam kategori. Untuk persamaan rumus interval adalah sebagai berikut.

$$I = 100 / \text{Jumlah Skor Likert}$$

Jumlah skor likert adalah jumlah skor yang digunakan, dalam penelitian ini adalah 4. Jadi $I = 100/4 = 25\%$, maka akan didapatkan kumpulan sebagai berikut.

Angka 0% - 24,99% = Buruk

Angka 25% - 49,99% = Kurang Baik

Angka 50% - 74,99% = Baik

Angka 75% - 100% = Sangat Baik

1. Perhitungan skala likert kategori kesesuaian

$$\text{Jumlah sangat setuju} = 18 \times 4 = 72$$

$$\text{Jumlah setuju} = 69 \times 3 = 207$$

$$\text{Jumlah tidak setuju} = 3 \times 2 = 6$$

$$\text{Jumlah sangat tidak setuju} = 0 \times 1 = 0$$

$$\text{Jumlah pemilih} = 90$$

$$\text{Rumus indeks \%} = 285 / 360 \times 100 = 79,16$$

2. Perhitungan skala likert kategori nilai

$$\text{Jumlah sangat setuju} = 11 \times 4 = 44$$

$$\text{Jumlah setuju} = 65 \times 3 = 195$$

$$\text{Jumlah tidak setuju} = 14 \times 2 = 28$$

$$\text{Jumlah sangat tidak setuju} = 0 \times 1 = 0$$

$$\text{Jumlah pemilih} = 90$$

$$\text{Rumus indeks \%} = 267 / 360 \times 100 = 74,16$$

3. Perhitungan skala likert kategori kemudahan

$$\text{Jumlah sangat setuju} = 16 \times 4 = 64$$

$$\text{Jumlah setuju} = 64 \times 3 = 192$$

$$\text{Jumlah tidak setuju} = 10 \times 2 = 20$$

$$\text{Jumlah sangat tidak setuju} = 0 \times 1 = 0$$

$$\text{Jumlah pemilih} = 90$$

$$\text{Rumus indeks \%} = 266 / 360 \times 100 = 73,88$$

4. Perhitungan skala likert kategori perasaan

$$\text{Jumlah sangat setuju} = 11 \times 4 = 44$$

$$\text{Jumlah setuju} = 67 \times 3 = 201$$

$$\text{Jumlah tidak setuju} = 12 \times 2 = 24$$

$$\text{Jumlah sangat tidak setuju} = 0 \times 1 = 0$$

$$\text{Jumlah pemilih} = 90$$

$$\text{Rumus indeks \%} = 267 / 360 \times 100 = 74,16$$

Berikut adalah tabel hasil perhitungan pengujian menggunakan skalan likert dan pengelompokannya sesuai kategori.

Tabel 6.7 Hasil pengujian koresponden

No	Kategori	Pertanyaan	Nilai	Keterangan
1	Kesesuaian	Apakah penggunaan <i>view controller</i> dalam bermain game bus simulator sudah sesuai dengan pergerakan kepala ketika melihat depan, menoleh ke kanan, dan ke kiri?	79,16%	Sangat Baik
2		Apakah penggunaan <i>view controller</i> ini sudah bagus dibandingkan dengan menggunakan <i>mouse</i> ?		
3		Apakah anda menyukai penggunaan <i>view controller</i> untuk bermain game Bus Simulator		
4	Nilai	Pada saat penggunaan <i>view controller</i> dalam bermain Bus Simulator, apakah <i>view controller</i> sudah stabil?	74,16%	Baik
5		Pada saat penggunaan <i>view controller</i> dalam bermain Bus Simulator, tingkat sensitifitas dari <i>view controller</i> tersebut sudah sesuai dengan batas sudut saat digunakan untuk menoleh ke kanan dan ke kiri		
6		Pada saat bermain game bus simulator, apakah <i>view controller</i> sudah akurat dalam penggunaannya?		
7	Kemudahan	Apakah dalam bermain game bus simulator dengan penggunaan <i>View Controller</i> ini mudah digunakan?	73,88	Baik

Tabel 6.7 Hasil pengujian koresponden (lanjutan)

No	Kategori	Pertanyaan	Nilai	Keterangan
8	Kemudahan	Pada saat memainkan game bus simulator dengan menggunakan <i>View Controller</i> , apakah mudah saat mengendalikan pergerakan menoleh ke kiri dan ke kanan?	73,38	Baik
9		apakah <i>view controller</i> sudah cocok untuk digunakan dalam bermain game bus simulator?		
10	Perasaan	Apakah anda senang saat memainkan game bus simulator dengan menggunakan <i>View Controller</i> ?	74,16	Baik
11		Apakah penggunaan <i>view controller</i> dalam bermain game bus simulator sudah baik saat digunakan?		
12		Apakah anda merasa nyaman saat menggunakan <i>View Controller</i> dalam bermain game bus simulator?		

Berdasarkan hasil dari pengujian terhadap *feedback* dari setiap *user*, pada kategori pengujian pertama yaitu korespondensi kesesuaian, pertanyaan pertama sebanyak 13,3% *user* sangat setuju, 83,3% setuju dan 3,3% tidak setuju, maka pertanyaan pertama ini dapat disimpulkan *user* bisa menggunakan *view controller* dengan sesuai pergerakan kepala dari *user*. Pada pertanyaan kedua sebanyak 20% *user* sangat setuju, 73,3% setuju dan 6,7% tidak setuju, maka dari pertanyaan kedua ini dapat disimpulkan bahwa *user* setuju bahwa penggunaan *view controller* lebih bagus dari pada penggunaan *mouse* dalam game bus simulator. Pada pertanyaan ketiga sebanyak 26,7% *user* sangat setuju dan 73,3% setuju, maka dari pertanyaan ketiga ini dapat disimpulkan bahwa *user* menyukai penggunaan *view controller* dalam bermain game bus simulator.

Pada kategori pengujian kedua yaitu korespondensi nilai, pertanyaan pertama sebanyak 13,3% *user* sangat setuju, 63,3% setuju dan 23,3% tidak setuju, maka dapat disimpulkan bahwa *user* dalam penggunaan *view controller* sudah stabil. Pada pertanyaan kedua sebanyak 10% *user* sangat setuju, 80% setuju dan 10% tidak setuju, maka dapat disimpulkan bahwa nilai dari sensitifitas dari *view controller* sudah sesuai dengan pergerakan *user*. Pada pertanyaan ketiga sebanyak 13,3% *user* sangat setuju, 73,3% setuju dan 13,3% tidak setuju, maka dari pertanyaan ketiga ini penggunaan *view controller* sudah akurat.

Pada kategori pengujian ketiga yaitu korespondensi kemudahan, pertanyaan pertama sebanyak 10% *user* sangat setuju, 80% setuju dan 10% tidak setuju, maka dapat disimpulkan bahwa *view controller* mudah digunakan. Pada pertanyaan kedua sebanyak 23,3% *user* sangat setuju, 63,3% setuju dan 13,3% tidak setuju, maka dapat disimpulkan bahwa *view controller* mudah dikendalikan. Pada pertanyaan ketiga sebanyak 20% *user* sangat setuju, 70% setuju dan 10%

tidak setuju, maka dapat disimpulkan game bus simulator cocok dimainkan dengan *view controller*.

Pada kategori pengujian keempat yaitu korespondensi perasaan, pertanyaan pertama sebanyak 13,3% *user* sangat setuju, 76,7% setuju dan 10% tidak setuju, maka dapat disimpulkan *user* senang dengan menggunakan *view controller*. Pada pertanyaan kedua sebanyak 13,3% *user* sangat setuju, 83,3% setuju dan 3,3% tidak setuju, maka dapat disimpulkan bahwa penggunaan *view controller* dalam bermain game bus simulator sudah baik digunakan. Pada pertanyaan ketiga sebanyak 10% *user* sangat setuju, 63,3% setuju dan 26,7% tidak setuju, maka dapat disimpulkan sebagian besar *user* nyaman saat menggunakan *view controller*.



BAB 7 KESIMPULAN

Pada bab ini akan dibahas tentang penarikan kesimpulan dari penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

7.1 Kesimpulan

Berdasarkan hasil dari perancangan, implementasi dan pengujian yang telah dilakukan, maka akan diambil beberapa kesimpulan sebagai berikut:

1. Sensor MPU9250 mampu membaca setiap perubahan posisi kepala secara akurat oleh *user*. Dikarenakan pada sensor tersebut merupakan gabungan dua sensor yaitu *accelerometer* dan *gyroscope*. Pergerakan yang dibaca oleh sistem adalah pergerakan kepala dengan menoleh ke kanan dan menoleh ke kiri.
2. Sistem yang dirancang mampu mengendalikan game bus simulator dengan menggunakan perubahan posisi kepala *user*. Pada perubahan kepala *user* saat menoleh ke kanan maupun ke kiri sistem dapat memberikan batas menoleh dari kursor, yang mana didalam game bus simulator kursor ada sudut maksimal yang menyesuaikan dengan *experience user* ketika menoleh ke kanan dan ke kiri. Dengan adanya batas sudut maksimal ini *user* mendapatkan *experience* bermain game dengan mudah saat menoleh ke kanan atau ke kiri kemudian kembali ke tengah.
3. Berdasarkan hasil pengujian koresponden yang dilakukan terhadap 30 orang *user*. Pada kategori kesesuaian, sistem mendapatkan nilai persentase sebesar 79,16% sehingga termasuk pada kategori sangat baik. Pada kategori nilai, sistem mendapatkan nilai persentase sebesar 74,16% sehingga termasuk dalam kategori baik. Pada kategori kemudahan sistem mendapatkan nilai persentase sebesar 73,88% sehingga termasuk dalam kategori baik. Pada kategori yang terakhir yaitu perasaan, sistem mendapatkan nilai persentase sebesar 74,16% sehingga termasuk dalam kategori baik. Dari nilai persentase yang didapatkan saat pengujian korespondensi disetiap kategori maka dapat disimpulkan secara keseluruhan sistem ini mampu mendapatkan rata-rata nilai BAIK. Dengan hasil ini maka penerapan *view controller* terhadap *user experience* pada game bus simulator sukses diimplementasikan.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem ini antara lain adalah:

1. Untuk pengembangan lebih lanjut, sistem dapat dikembangkan dengan menggunakan sensor yang lebih akurat untuk menghasilkan *experience* yang lebih baik.
2. Untuk pengembangan lebih lanjut, peneliti dapat menyederhanakan bentuk perangkat keras.

3. Untuk pengembangan lebih lanjut, peneliti supaya dapat menggunakan kode program yang lebih sederhana dan padat dan mudah serta bisa menambahkan metode-metode lain yang relevan.



DAFTAR PUSTAKA

- AJIE, 2015. Komunikasi Serial Sinkron I2C dengan Arduino. [Online] Tersedia di: <
<http://saptaji.com/2015/07/24/komunikasi-serial-sinkron-i2ciictwi-dengan-arduino/> > [27 Januari 2018].
- Arduino, 2015. MPU9250 Accelerometer. [Online] Tersedia di:
 <<http://playground.arduino.cc/Main/MMA7455>> [Diakses 27 Januari 2018].
- Arduino, 2015. *Library Mouse*. [Online] Tersedia di: <
<https://www.arduino.cc/reference/en/language/functions/usb/mouse/>>
 [Diakses 29 Maret 2018].
- Datasheet MPU – 60XX,. 2013. Tersedia di: < <https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf> > [Diakses 04 Februari 2018]
- Ervanda, Virza Audy. “Pengembangan Sistem Deteksi Gerakan Kepala Sebagai Kontrol Pergerakan Kursi Roda Berbasis *Embedded Sistem*.” Skripsi. Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya (UB). (2017)
- Khalif, Idham. “Pengembangan Sistem Penghitungan langkah Kaki hemat Daya Berbasis Wemos D1 *Mini*.” Skripsi. Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya (UB). (2017)
- Liem, Yuliana Kathina Hatta. “Rancang Bangun Kursi Roda Elektrik Menggunakan Perintah Suara Berbasis Aplikasi Android.” Skripsi. Teknik Elektro, Fakultas Teknik Industri, Institut Teknologi Sepuluh Nopember (ITS). (2012)
- MPU-6050, TDK Invensense. “MPU-6050 Six-Axis (Gyro + Accelerometer) MEMS MotionTracking™ Devices.” URL:
<https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>
- Purwoko, Aji. 2018. “Manfaat Bermain Game Bagi Kesehatan” tersedia di: <
<https://jalantikus.com/tips/manfaat-bermain-game-bagi-kesehatan/>>
 [Diakses 20 Mei 2018]
- Studio, Still Alive 2016. Bus Simulator [online] Tersedia di:
http://store.steampowered.com/app/324310/Bus_Simulator_16/ [Diakses 1 Februari 2018].
- Vermelen,Arnold P.O.S. “*User Experience Evaluation Methods: Current State And Development Needs*”. Proceeding : Nordichi. Oktober 2010